



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: Desarrollo de un sistema de reconocimiento facial

AUTOR: Abdón Alejandro Vivas Imparato

TITULACIÓN: Grado en Ingeniería de Sonido e Imagen

TUTOR (o Director en su caso): Danilo Simón Zorita

DEPARTAMENTO: DIAC

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: Juana María Gutiérrez Arriola

VOCAL: Danilo Simón Zorita

SECRETARIO: Alfonso Martín Marcos

Fecha de lectura:

Calificación:

El Secretario,

A Inés, mis abuelas y mi dorada Venezuela

AGRADECIMIENTOS

A Zully e Inés, por toda su ayuda, apoyo y preocupación.

A mis padres, por todo su esfuerzo y trabajo duro para que yo llegara hasta aquí.

A Danilo, por su espléndido trabajo como tutor.

A mi hermano Diego, por haber escuchado mis propuestas y avances.

A Champi, por proporcionarme material de trabajo.

A Javi jebi, “Chispa Chispa Chispa” y Christian por recordarme detalles importantes de formato.

A mis compañeros de Kulturales, por su compañía durante todo el grado y por todos esos momentos de alegría (la Figura 9.16 va dedicada a todos ustedes). ¡Felicidades j·ingenieros!

A mi tío Tony Eblén por sus prometedoras propuestas antes de decantarme por este proyecto.

Por último, y no menos importante, a Nikola Tesla, por toda su investigación y aportes al mundo de la telecomunicación.

RESUMEN

El objetivo principal alrededor del cual se desenvuelve este proyecto es el desarrollo de un sistema de reconocimiento facial. Entre sus objetivos específicos se encuentran: realizar una primera aproximación sobre las técnicas de reconocimiento facial existentes en la actualidad, elegir una aplicación donde pueda ser útil el reconocimiento facial, diseñar y desarrollar un programa en MATLAB que lleve a cabo la función de reconocimiento facial, y evaluar el funcionamiento del sistema desarrollado.

Este documento se encuentra dividido en cuatro partes: INTRODUCCIÓN, MARCO TEÓRICO, IMPLEMENTACIÓN, y RESULTADOS, CONCLUSIONES Y LÍNEAS FUTURAS.

En la primera parte, se hace una introducción relativa a la actualidad del reconocimiento facial y se comenta brevemente sobre las técnicas existentes para desarrollar un sistema biométrico de este tipo. En ella se justifican también aquellas técnicas que acabaron formando parte de la implementación.

En la segunda parte, el marco teórico, se explica la estructura general que tiene un sistema de reconocimiento biométrico, así como sus modos de funcionamiento, y las tasas de error utilizadas para evaluar y comparar su rendimiento. Así mismo, se lleva a cabo una descripción más profunda sobre los conceptos y métodos utilizados para efectuar la detección y reconocimiento facial en la tercera parte del proyecto.

La tercera parte abarca una descripción detallada de la solución propuesta. En ella se explica el diseño, características y aplicación de la implementación; que trata de un programa elaborado en MATLAB con interfaz gráfica, y que utiliza cuatro sistemas de reconocimiento facial, basados cada uno en diferentes técnicas: Análisis por componentes principales, análisis lineal discriminante, *wavelets* de Gabor, y emparejamiento de grafos elásticos. El programa ofrece además la capacidad de crear y editar una propia base de datos con etiquetas, dándole aplicación directa sobre el tema que se trata. Se proponen además una serie de características con el objetivo de ampliar y mejorar las funcionalidades del programa diseñado. Dentro de dichas características destaca la propuesta de un modo de verificación híbrido aplicable a cualquier rama de la biometría y un programa de evaluación capaz de medir, graficar, y comparar las configuraciones de cada uno de los sistemas de reconocimiento implementados. Otra característica destacable es la herramienta programada para la creación de grafos personalizados y generación de modelos, aplicable a reconocimiento de objetos en general.

En la cuarta y última parte, se presentan al principio los resultados obtenidos. En ellos se contemplan y analizan las comparaciones entre las distintas configuraciones de los sistemas de reconocimiento implementados para diferentes bases de datos (una de ellas formada con imágenes con condiciones de adquisición no controladas). También se miden las tasas de error del modo de verificación híbrido propuesto. Finalmente, se extraen conclusiones, y se proponen líneas futuras de investigación.

ABSTRACT

The main goal of this project is to develop a facial recognition system. To meet this end, it was necessary to accomplish a series of specific objectives, which were: researching on the existing face recognition technics nowadays, choosing an application where face recognition might be useful, design and develop a face recognition system using MATLAB, and measure the performance of the implemented system.

This document is divided into four parts: INTRODUCTION, THEORETICAL FRAMEWORK, IMPLEMENTATION, and RESULTS, CONCLUSIONS AND FUTURE RESEARCH STUDIES.

In the first part, an introduction is made in relation to facial recognition nowadays, and the techniques used to develop a biometric system of this kind. Furthermore, the techniques chosen to be part of the implementation are justified.

In the second part, the general structure and the two basic modes of a biometric system are explained. The error rates used to evaluate and compare the performance of a biometric system are explained as well. Moreover, a description of the concepts and methods used to detect and recognize faces in the third part is made.

The design, characteristics, and applications of the systems put into practice are explained in the third part. The implementation consists in developing a program with graphical user interface made in MATLAB. This program uses four face recognition systems, each of them based on a different technique: Principal Component Analysis (PCA), Fisher's Linear Discriminant (FLD), Gabor wavelets, and Elastic Graph Matching (EGM). In addition, with this implementation it is possible to create and edit one's tagged database, giving it a direct application. Also, a group of characteristics are proposed to enhance the functionalities of the program designed. Among these characteristics, three of them should be emphasized in this summary: A proposal of an hybrid verification mode of a biometric system; and an evaluation program capable of measuring, plotting curves, and comparing different configurations of each implemented recognition system; and a tool programmed to create personalized graphs and models (tagged graph associated to an image of a person), which can be used generally in object recognition.

In the fourth and last part of the project, the results of the comparisons between different configurations of the systems implemented are shown for three databases (One of them created with pictures taken under non-controlled environments). The error rates of the proposed hybrid verification mode are measured as well. Finally, conclusions are extracted and future research studies are proposed.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	III
RESUMEN.....	V
ABSTRACT.....	VII
ÍNDICE GENERAL.....	IX
LISTA DE ACRÓNIMOS.....	XI
PARTE I: INTRODUCCIÓN.....	1
1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	2
1.2. METODOLOGÍA.....	3
PARTE II: MARCO TEÓRICO.....	5
2. SISTEMAS BIOMÉTRICOS.....	5
2.1. <i>Arquitectura de un sistema biométrico</i>	5
2.2. <i>Modos de funcionamiento</i>	6
2.3. <i>Evaluación de los modos de funcionamiento</i>	6
3. SISTEMAS BIOMÉTRICOS DE RECONOCIMIENTO FACIAL.....	11
3.1. <i>Condiciones de captura</i>	11
3.2. <i>Detección y segmentación</i>	11
3.3. <i>Pre-procesado</i>	12
3.4. <i>Extracción de características y clasificación</i>	13
4. DETECCIÓN FACIAL.....	15
4.1. <i>El algoritmo AdaBoost</i>	15
4.2. <i>Detección facial usando el algoritmo de Viola-Jones</i>	18
4.2.1. <i>Características</i>	18
4.2.2. <i>Imagen integral</i>	20
4.2.3. <i>Clasificadores débiles y variante del algoritmo Adaboost</i>	21
4.2.4. <i>Clasificadores en cascada</i>	25
4.2.5. <i>Otras consideraciones</i>	26
5. ANÁLISIS POR COMPONENTES PRINCIPALES (PCA).....	27
5.1. <i>Conceptos básicos y nomenclatura</i>	27
5.2. <i>El análisis por componentes principales</i>	29
5.3. <i>Reconocimiento facial utilizando eigenfaces</i>	32
6. ANÁLISIS LINEAL DISCRIMINANTE (LDA).....	39
6.1. <i>Reconocimiento facial usando fisherfaces</i>	44
7. <i>WAVELETS</i> DE GABOR.....	47
7.1. <i>El análisis multirresolución y las wavelets en general</i>	47
7.2. <i>Filtros de Gabor</i>	51
7.3. <i>Las wavelets de Gabor</i>	54
7.4. <i>Reconocimiento facial basado en wavelets de Gabor</i>	58
8. EMPAREJAMIENTO POR GRAFOS ELÁSTICOS (EGM).....	63
8.1. <i>Etapa de entrenamiento</i>	64
8.2. <i>Etapa de clasificación</i>	65
8.3. <i>Elastic Bunch Graph Matching</i>	67
PARTE III: IMPLEMENTACIÓN.....	69
9. DESCRIPCIÓN GENERAL DE LA IMPLEMENTACIÓN.....	69
9.1. <i>Detección automática</i>	70
9.2. <i>Estructura de diseño</i>	71
9.3. <i>Gestión de la base de datos y propiedades generales</i>	74

9.4.	<i>Historial de clasificación.....</i>	76
9.5.	<i>Tipo de Umbral y discriminación del sistema ante individuos externos a la base de datos</i>	77
9.6.	<i>Sistema de reconocimiento facial basado en PCA</i>	79
9.7.	<i>Sistema de reconocimiento facial basado en LDA</i>	81
9.8.	<i>Sistema de reconocimiento facial basado en wavelets de Gabor</i>	82
9.9.	<i>Sistema de reconocimiento facial basado en EGM</i>	86
9.10.	<i>Ventanas de visualización</i>	91
9.11.	<i>Reconocimiento facial en modo identificación</i>	92
9.12.	<i>Reconocimiento facial en modo verificación</i>	95
9.13.	<i>Propuesta de un modo híbrido de clasificación</i>	96
9.14.	<i>Implementación de un programa de evaluación.....</i>	96
PARTE IV: RESULTADOS, CONCLUSIONES Y LÍNEAS FUTURAS		103
10.	RESULTADOS	103
10.1.	<i>Curvas de evaluación sobre la base de datos de Yale University</i>	104
10.2.	<i>Base de datos del Centro Universitário da FEI</i>	114
10.3.	<i>Base de datos personal no-controlada</i>	122
10.4.	<i>Comparación entre el modo de verificación tradicional y el modo de verificación híbrido propuesto</i>	128
10.5.	<i>Evaluación de sistema de reconocimiento facial basado en EGM.....</i>	133
11.	CONCLUSIONES.....	137
12.	PRINCIPALES CONTRIBUCIONES	139
13.	LÍNEAS FUTURAS DE INVESTIGACIÓN	141
REFERENCIAS		143
ÍNDICE DE FIGURAS.....		147
ÍNDICE DE TABLAS		151

ANEXO

LISTA DE ACRÓNIMOS

CAR:	<i>Correct Acceptance Rate</i> ; Tasa de correcta aceptación.
CRR:	<i>Correct Rejection Rate</i> ; Tasa de correcto rechazo.
DET:	<i>Detection Error Tradeoff</i> .
EBGM:	<i>Elastic Bunch Graph Matching</i> .
EER:	<i>Equal Error Rate</i> ; Tasa de igual error.
EGM:	<i>Elastic Graph Matching</i> ; Emparejamiento de grafos elásticos.
FAR:	<i>False Acceptance Rate</i> ; Tasa de falsa aceptación.
FBG:	<i>Face Bunch Graph</i> .
FLD:	<i>Fisher's Linear Discriminant</i> ; Análisis lineal discriminante de Fisher.
FRR:	<i>False Rejection Rate</i> ; Tasa de falso rechazo.
LDA:	<i>Linear Discriminant Analysis</i> ; Análisis lineal discriminante.
PCA:	<i>Principal Component Analysis</i> ; Análisis por componentes principales.
ROC:	<i>Receiver Operating Characteristics</i> .
STFT:	<i>Short Time Fourier Transform</i> ; Transformada localizada de Fourier.

PARTE I: INTRODUCCIÓN

1. Introducción

Se define como **biometría** al estudio de la automatización del reconocimiento de un individuo a partir rasgos conductuales o fisiológicos. La importancia de la biometría tiene su origen en la ventaja que supone su uso en ciertas aplicaciones, de entre las cuales se encuentran: control de acceso a una información, sistema, o sitio; identificación civil, controles de vigilancia, identificación criminal, etc. Unas de las principales ventajas de la biometría son la comodidad y seguridad que ésta presenta ante los sistemas tradicionales de acceso, puesto que no haría falta llevar consigo tarjetas de identificación o llaves, preocuparse por el robo o pérdida de éstas, ni recordar contraseñas complicadas. Otra ventaja destacable del uso de la biometría es la identificación de un individuo dentro de un conjunto amplio de personas. Entre los sistemas biométricos se encuentran el reconocimiento de iris, huella dactilar, huella palmar voz, firma escrita, reconocimiento facial, entre otros.

Uno de los sistemas biométricos que ha tenido mayor repercusión a lo largo de los años es el reconocimiento facial, y a pesar de que actualmente existen otros métodos de identificación biométrica muy fiables (como el reconocimiento de huellas dactilares o de iris), éste se sigue siendo un área de investigación activa por su naturaleza amigable.

Dicho esto, se eligió el reconocimiento facial como objetivo de estudio de este proyecto, con el fin de aportar ideas que puedan servir de utilidad en esta área de la biometría.

Las técnicas de reconocimiento facial se pueden dividir en dos familias: Las basadas en apariencia, y las basadas en modelos. Las técnicas basadas en apariencia manejan cada imagen de un rostro como una representación de un punto en un subespacio vectorial, mientras que las técnicas basadas en modelos pretenden construir un modelo capaz de capturar variaciones faciales. A su vez, las técnicas basadas en apariencia pueden ser divididas en lineales y no lineales, y las técnicas basadas en modelos pueden ser divididas en modelos 2D y modelos 3D. [1]

Las técnicas lineales basadas en apariencia, como su nombre indica, proyectan las imágenes sobre un subespacio vectorial lineal, y entre ellas se puede encontrar: Análisis por componentes principales, análisis lineal independiente, análisis lineal discriminante, entre otras. Por otro lado, las técnicas no lineales utilizan métodos de mapeado no lineal de los datos, como por ejemplo los métodos Kernel, Isomap, o *Locally Linear Embedding*.

Al igual que las técnicas basadas en apariencia, las técnicas basadas en modelos se pueden dividir en técnicas basadas en modelos 2D (emparejamiento de grafos elásticos,

modelos de apariencia activa, etc.), y técnicas basadas en modelos 3D (grafos de tres dimensiones).

Los sistemas de reconocimiento facial basados en apariencia resultan más rápidos que los sistemas basados en modelos, menos complejos y no requieren un conocimiento previo de las imágenes. Sin embargo, los sistemas de reconocimiento facial basados en modelos son más robustos frente a cambios de orientación o expresión de la cara, y presentan menos inconvenientes en cuanto a cambios en la iluminación. [2]

Como se puede deducir de lo anteriormente comentado, hoy en día existe un número muy alto de técnicas de reconocimiento facial (algunas de ellas más complejas que otras), por lo que resulta evidente que hay que tomar una decisión cuando se desea diseñar un sistema de reconocimiento facial.

En consecuencia, al llevar a cabo este proyecto se optó por profundizar sobre cuatro técnicas, elegidas por su relación con los conocimientos adquiridos durante el grado en Ingeniería de Sonido e Imagen de la Universidad Politécnica de Madrid. Por lo tanto, para poder entender el contenido de este proyecto, es imprescindible conocer información sobre conceptos relacionados con operaciones sobre espacios vectoriales, análisis estadístico de datos multivariados, procesamiento digital de la señal, transformada de Fourier, entre otros. Las técnicas escogidas fueron: Análisis por componentes principales, análisis lineal discriminante, *wavelets* de Gabor, y emparejamiento de grafos elásticos.

1.1.Objetivos

Objetivo general:

Diseñar e implementar un sistema de reconocimiento facial.

Objetivos específicos:

1. Realizar una primera aproximación a las diferentes técnicas de reconocimiento facial existentes.
2. Escoger una situación donde sea necesario el desarrollo de un sistema de reconocimiento facial e implementar las técnicas que se consideren más apropiadas.
3. Evaluar el funcionamiento de las técnicas implementadas.
4. Aprender a diseñar interfaces gráficas con MATLAB y basar el desarrollo del sistema en este lenguaje de alto nivel.
5. Proponer soluciones a los problemas que surjan durante el desarrollo del proyecto.
6. Generar aportaciones provechosas para futuras investigaciones en reconocimiento facial.
7. Analizar los resultados obtenidos y obtener conclusiones.

1.2. Metodología

La metodología de trabajo se compuso de tres fases claramente identificadas: La primera fue la fase de investigación, la cual se centró en consultas a publicaciones en revistas técnicas y científicas, bibliografía específica y enlaces web; la segunda fue la fase de desarrollo y toma de decisiones, en la cual se diseñó e implementó el programa utilizando interfaces gráficas de MATLAB; y la última, la fase de evaluación y análisis de los resultados obtenidos, donde se compararon los sistemas de reconocimiento implementados, se obtuvieron conclusiones, y se propusieron líneas futuras de investigación.

PARTE II: MARCO TEÓRICO

2. Sistemas biométricos

Un sistema biométrico es un sistema de clasificación de individuos donde cada individuo es reconocido automáticamente a partir de sus características fisiológicas o de comportamiento. Entre las características fisiológicas se pueden encontrar el rostro, huella dactilar, huella palmar, iris, etc. y representan aquellas características que varían muy poco a lo largo del tiempo para una persona. Por otro lado, entre las características de comportamiento se encuentran la forma de teclear (fuerza, velocidad, etc.), firma escrita, voz, entre otras, y representan características dinámicas de una persona. [3]

2.1. Arquitectura de un sistema biométrico

En la Figura 2.1 se puede observar el diagrama de bloques genérico de un sistema biométrico.

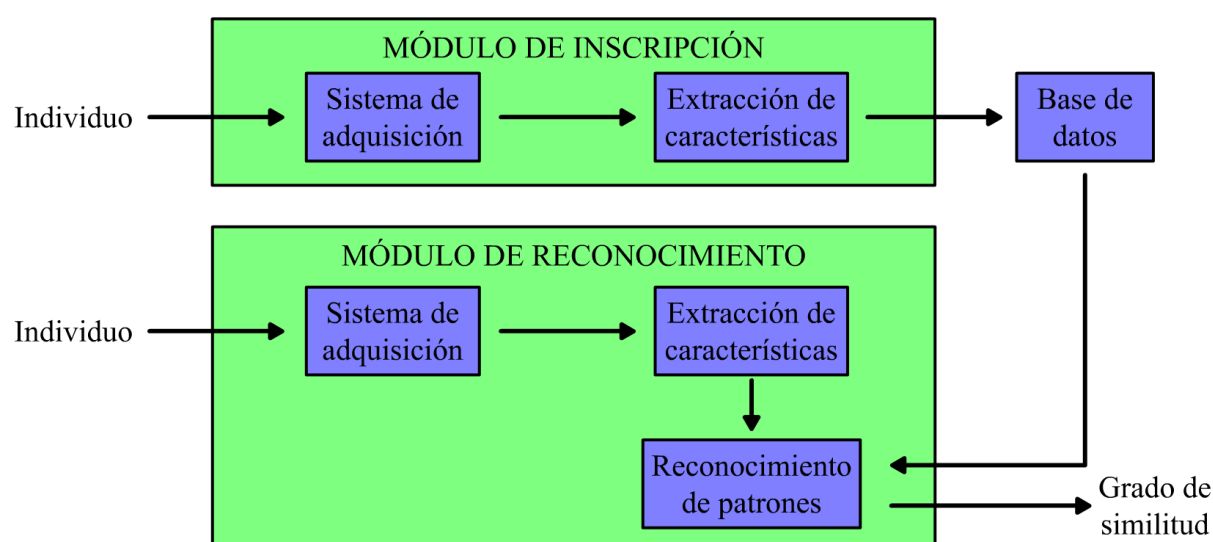


Figura 2.1. Diagrama de bloques genérico de un sistema biométrico.

Como se puede observar, existen dos módulos claramente identificados: El módulo de inscripción y el módulo de reconocimiento. El primero de ellos es donde se lleva a cabo la incorporación de un individuo a la base de datos (entrenamiento) y el segundo es donde se efectúa el reconocimiento a partir de los datos recolectados por el módulo anterior.

El sistema de adquisición está formado por un transductor que genera una señal a partir de la fisiología o comportamiento del individuo para enviarla al bloque de extracción de características, donde se hacen los cálculos y mediciones necesarias para formar un **patrón**, que dependiendo del módulo, se almacena en la base de datos (módulo de inscripción) o se envía a un clasificador (bloque de reconocimiento de patrones en el

módulo de reconocimiento) cuya función es comparar el patrón medido con los almacenados en la base de datos para tomar una decisión.

2.2. Modos de funcionamiento

Dentro de los sistemas biométricos existen dos modos de funcionamiento:

Modo identificación: En este modo el sistema determina automáticamente la identidad del individuo si éste se encuentra almacenado en la base de datos. Muy a menudo se diseña el sistema para presentar una lista en orden de similitud de los individuos cuyos patrones se asemejan más al patrón medido.

Dentro de este modo de funcionamiento existen además dos tipos: identificación en conjunto cerrado, donde el sistema se restringe a identificar al individuo según los almacenados en la base de datos; e identificación en conjunto abierto, donde cabe la posibilidad de que no exista correspondencia con ningún individuo de la base de datos y el sistema lo identifique como desconocido. Un ejemplo del modo identificación pueden ser los sistemas de reconocimiento facial o de huellas dactilares que se utilizan en investigaciones criminales.

Modo verificación: El objetivo de este modo de funcionamiento es determinar si el individuo que accede al sistema es quien dice ser. Por lo tanto, el usuario debe proporcionarle al sistema cuál es su identidad antes de empezar el procedimiento de adquisición y comprobación.

Para este modo existen 4 posibles resultados: **Correcta aceptación, correcto rechazo, falsa aceptación y falso rechazo**. Como sus nombres indican, una situación de correcta aceptación se da cuando un individuo legítimo es clasificado como tal, un correcto rechazo se da cuando un individuo (impostor) se hace pasar por otro y consecuentemente es rechazado por el sistema, una falsa aceptación se da cuando un impostor es aceptado como legítimo y un falso rechazo se da cuando un individuo legítimo es rechazado como impostor

2.3. Evaluación de los modos de funcionamiento

Modo verificación:

Si se llama H_0 a la hipótesis en la que un individuo es clasificado como impostor, H_1 a la hipótesis contraria (es clasificado como legítimo), D_0 a la decisión del sistema cuando el individuo es clasificado como impostor, y D_1 es la decisión contraria; se pueden definir las siguientes tasas para medir el rendimiento del sistema:

- **Tasa de falsa aceptación (FAR, por sus siglas en inglés):** Es la probabilidad de que un individuo impostor sea clasificado como legítimo.

$$FAR = P(D_1/H_0) \quad (2.1)$$

- **Tasa de falso rechazo (FRR, por sus siglas en inglés):** Es la probabilidad de que un individuo legítimo sea clasificado como impostor.

$$FRR = P(D_0/H_1) \quad (2.2)$$

- **Tasa de correcta aceptación (CAR, por sus siglas en inglés):** Es la probabilidad de que un individuo legítimo sea clasificado como tal.

$$CAR = P(D_1/H_1) \quad (2.3)$$

- **Tasa de correcto rechazo (CRR, por sus siglas en inglés):** Es la probabilidad de que un impostor sea clasificado como tal.

$$CRR = P(D_0/H_0) \quad (2.4)$$

Dichas tasas pueden ser representadas de forma gráfica utilizando las funciones de densidad de probabilidad de individuos legítimos e impostores. Estas curvas se obtienen midiendo un patrón de entrada de cada individuo con su patrón correspondiente en la base de datos (función de densidad de individuos legítimos) y con los patrones almacenados de los otros individuos (función de densidad de impostores). Cada comparación producirá una similitud s , resultando unas curvas con una forma parecida a las de la Figura 2.2.

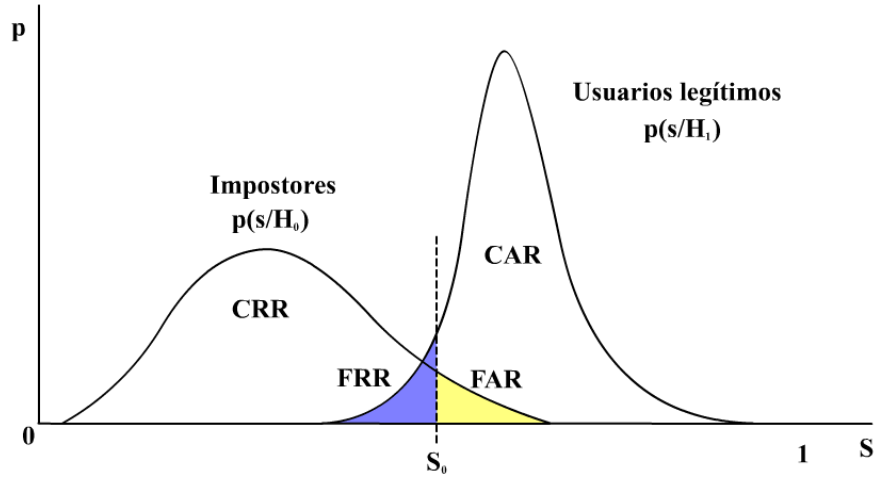


Figura 2.2. Funciones densidad de probabilidad de individuos legítimos e impostores.

Como se puede observar en la figura, las funciones de densidad de probabilidad de individuos legítimos e impostores se solapan en una zona donde es necesario trazar un umbral S_0 para que el sistema tome una decisión. A partir de dicho umbral se forman las zonas de falsa aceptación, falso rechazo, correcta aceptación y correcto rechazo. En consecuencia, las tasas comentadas anteriormente quedan definidas como:

$$FAR = \int_{S_0}^1 p(s/H_0) ds \quad (2.5)$$

$$FRR = \int_0^{S_0} p(s/H_1) ds \quad (2.6)$$

$$CAR = 1 - FRR \quad (2.7)$$

$$CRR = 1 - FAR \quad (2.8)$$

Si se grafican las tasas de falsa aceptación y falso rechazo en función del umbral decidido (Figura 2.3), al punto de cruce entre ambas curvas se le denomina **tasa de igual error** (EER, por sus siglas en inglés).

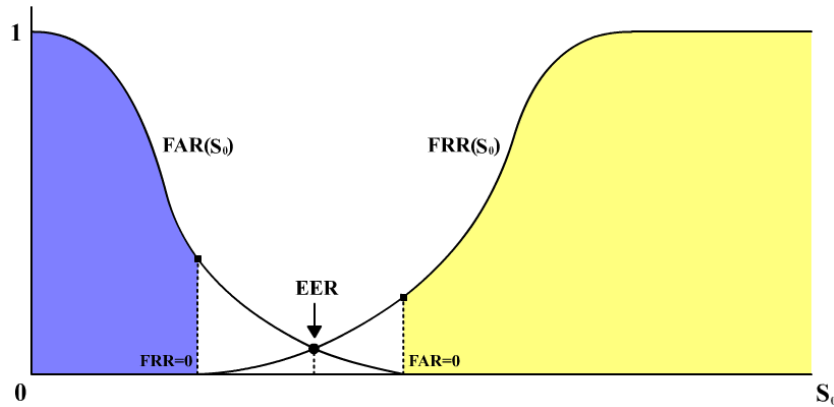


Figura 2.3. Tasas de falsa aceptación y falso rechazo en función del umbral de decisión.

Existe una gráfica que representa la relación entre las tasas de falsa aceptación y falso rechazo denominada ROC (por sus siglas en inglés, *Receiver Operator Characteristic*), donde el eje de las ordenadas representa la FAR y el de las abscisas el FRR. Algunos autores también suelen representar la ROC como la CAR en función de la FAR [4]. En la Figura 2.4 se puede observar que dentro de una ROC se puede trazar además una línea recta con los valores posibles que puede tener el EER y así comparar distintos sistemas con una sola gráfica. Un sistema de clasificación tiene mejor rendimiento cuanto más empinada esté su curva ROC hacia el origen, es decir, cuanto menor sea su EER.

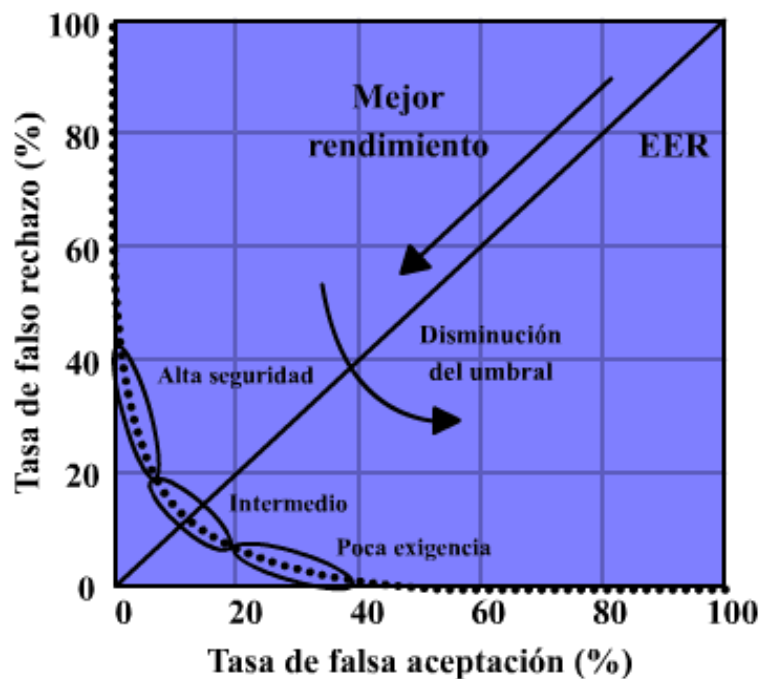


Figura 2.4. Curva ROC.

Otra manera de representar gráficamente la evaluación de un sistema en modo verificación es mediante la curva DET (*Detection Error Trade-off*). Esta última curva representa también la FRR en función de la FAR, pero se aplica una transformación normal a los ejes [5], aunque también suelen representarse en forma logarítmica simplemente. Como resultado, la curva que antes se observaba en la ROC, tiende más a una línea recta (Figura 2.5).



Figura 2.5. Curva DET.

Modo identificación:

De forma análoga al modo verificación, en el modo identificación se pueden definir unas tasas de Falsa No-coincidencia (equivalente a la tasa de falso rechazo en el modo verificación), y de Falsa Coincidencia (equivalente a la de falsa aceptación en el modo verificación). [6]

También es importante tomar en cuenta que normalmente el tiempo de respuesta de un sistema en modo de identificación suele ser mucho más lento que el de uno en modo verificación. Esto se debe a que no es lo mismo comparar con un solo individuo de la base de datos (verificación) para tomar una decisión, que comparar con todos los individuos de la base de datos (identificación).

3. Sistemas biométricos de reconocimiento facial

El diagrama de bloques general de los procedimientos que deben llevarse a cabo para realizar un reconocimiento facial a partir de una imagen corresponde con el de la Figura 3.1.

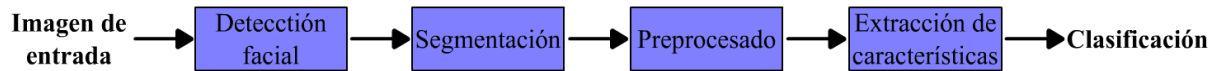


Figura 3.1. Diagrama de bloques general de un sistema de reconocimiento facial.

Es importante no confundir los términos **detección facial** y **reconocimiento facial**. El primero de ellos se refiere al proceso de localización de un rostro en una imagen y el segundo se refiere a la identificación o verificación de la identidad un individuo a partir una imagen de su rostro.

3.1. Condiciones de captura

Ambiente controlado: Se dice que la imagen de una cara fue capturada en un ambiente controlado cuando se fijan las características de captura para hacerla más favorable al reconocimiento. Por ejemplo: el tipo de fondo (normalmente uniforme), sin ningún otro objeto en la imagen, la iluminación del sitio donde se realiza la captura, o incluso la distancia de la cámara al rostro.

Ambiente no controlado: Es el contrario al ambiente controlado. en este caso no se garantiza ninguna característica favorable de captura, pudiendo tener la cara cualquier tamaño dentro de la imagen, estar ocluida por otros objeto, tener poca iluminación, la dirección de la luz puede ser desfavorable, puede haber otros objetos en la imagen o la imagen puede tener cualquier tipo de textura de fondo.

3.2. Detección y segmentación

La detección consiste en reconocer en qué zona de la imagen se encuentra una cara, dando lugar a la segmentación (extracción de dicho rostro de la imagen).

En el caso de poseer una base de datos de imágenes con solamente caras o que la imagen del individuo a clasificar ya esté limitada a la cara, no es necesaria la detección, pero de lo contrario es un paso fundamental. Existen varios métodos de detección facial:

Métodos basados en ambiente controlado:

Para imágenes de rostros con ambiente controlado suele ser suficiente técnicas sencillas de detección de borde.

Métodos basados en el color de piel:

Este tipo de métodos utiliza la crominancia de la imagen para detectar en qué zona de la imagen se encuentra una cara en función de los colores típicos de piel. El

entrenamiento suele basarse en muestras de piel y normalmente el método se combina con otros para poder decidir si el color detectado es efectivamente una cara (podría ser otra parte del cuerpo u otro objeto con colores semejantes a los de la piel). La mayor dificultad que tienen este tipo de métodos es que el rango existente de colores de piel es muy amplio, sin mencionar que varía mucho dependiendo de la iluminación, cámara utilizada, bronceados, etc. [7] [8]

Métodos basados en movimiento:

Utilizan una entrada de vídeo para la detección. Se basan en el hecho de que en un vídeo grabado con una cámara fija existen pocos objetos que se mueven, y entre ellos las caras. También pueden haber sido grabados con una cámara en movimiento, pero precisa de algoritmos un poco más complejos. Al igual que los métodos basados en color de piel, también suelen combinarse con otros métodos para comprobar que el objeto que se está moviendo es un rostro. [9]

Métodos basados en plantillas:

Este tipo de métodos consisten en hallar una plantilla o modelo que pueda describir con certeza un rostro. Por ejemplo, el modelo de una cara puede ser descrito por su contorno o por su simetría [10]. Otros métodos utilizan la similitud del contorno de un rostro con una elipse [11]. También puede dividirse la cara e identificar independientemente los ojos, la nariz o la boca.

Estos métodos suelen ser muy limitados y dejan mucho que desear en cuanto a resultados se refiere. Son muy sensibles al tamaño de la cara, pose, o cambios de iluminación. No obstante, existen otros métodos basados en plantillas deformables que consiguen sobrepasar estos problemas [12].

Otros métodos:

Existen métodos basados en *boosting*, que utilizan los resultados de numerosos clasificadores simples y los ponderan generando un clasificador fuerte. Uno de los métodos actuales más eficiente de detección facial (Viola-Jones) utiliza *boosting* adaptativo, y es el método de detección facial aplicado en este proyecto, por lo que se profundizará sobre él más adelante.

También existen otros métodos basados en el aspecto, que capturan durante una etapa de entrenamiento las variaciones representativas del aspecto final de un rostro [9] [13].

3.3. Pre-procesado

Antes de realizar una tarea de reconocimiento es imprescindible hacer un pre-procesado a las muestras recolectadas para favorecer la clasificación. Para el caso del reconocimiento facial se podría realizar algún tratamiento de imagen que reduzca la

influencia de los cambios de iluminación dentro de una misma imagen o la diferencia de luminancia que existe entre diferentes imágenes, por ejemplo.

3.4. Extracción de características y clasificación

Existen numerosas técnicas para extraer las características de la imagen de un rostro. En este proyecto, y por su relación con los conceptos aprendidos durante el grado, se estudiarán de forma más profunda las técnicas de análisis por componentes principales, análisis lineal discriminante y *wavelets* de Gabor. Otras técnicas de extracción de características y clasificación pueden ser: *Local Binary Patterns*, técnicas basadas en plantillas, redes neuronales artificiales, *Support Vector Machines*, etc.

Al conjunto de características extraídas de una muestra en cualquier sistema de clasificación se le denomina **patrón** y en función de los patrones almacenados en la etapa de entrenamiento se decide una regla o función que determine la clase de un patrón de entrada en la etapa de clasificación.

4. Detección facial

El proceso de localización de un rostro en una imagen no deja de ser una tarea de clasificación de dos clases: Caras y No-caras.

Como ya se ha comentado anteriormente, existen numerosos métodos de detección facial basados en diferentes tipos de técnicas. No obstante, el objetivo de este proyecto no es profundizar sobre las diferentes técnicas de detección facial, ya que éste es simplemente un primer paso a seguir para conseguir el reconocimiento facial de un individuo y en algunos casos puede que incluso no sea necesaria (ambiente controlado, imágenes previamente recortadas, entre otros). Por otro lado, tampoco se ha de ignorar la detección, puesto que aun así forma parte de la mayoría de los sistemas de reconocimiento facial (y este proyecto no es la excepción). Por lo tanto, y después de una investigación medianamente superficial, se optó por profundizar sobre uno de los métodos más eficientes utilizados en la actualidad, reconocido sobre todo por la relación entre su rapidez y efectividad: El algoritmo de Viola-Jones [14].

Antes de empezar, es fundamental comentar que el algoritmo de Viola-Jones utiliza una variante de un algoritmo muy conocido de clasificación denominado *AdaBoost*, que a su vez es un algoritmo de *Boosting*, pero ¿Qué es el *Boosting*?

El *Boosting* es un algoritmo meta-heurístico¹ que se basa en la siguiente pregunta: ¿Puede un conjunto de **clasificadores débiles**² unificarse para crear un **clasificador fuerte**? [15]

A pesar de que el *Boosting* no tiene restricciones, la gran mayoría de los algoritmos de *Boosting* se basa en entrenar clasificadores débiles de manera iterativa con respecto a una distribución de los datos y suma sus resultados de manera ponderada para crear un clasificador fuerte. Se puede decir que es un algoritmo de **aprendizaje**.

4.1. El algoritmo *AdaBoost*

Su nombre proviene del inglés *Adaptive Boosting* y significa ***Boosting* adaptativo**. Se trata de un algoritmo propuesto por Freund y Schapire [16] que consigue el aprendizaje a partir de la modificación de la distribución de las muestras al finalizar cada iteración.

¹ Un algoritmo meta-heurístico se refiere a aquel que está diseñado para encontrar, generar, o seleccionar una heurística que proporcione una solución lo suficientemente buena para un problema de optimización. En el entorno de la informática, se les califica como heurísticos a los algoritmos diseñados para encontrar una solución óptima, aunque ésta pueda ser errónea en la minoría de los casos. Generalmente se utilizan cuando no existe una solución óptima bajo la existencia de ciertas restricciones (tiempo, espacio, etc.).

² Un clasificador débil se define como aquel que clasifica mejor que una clasificación aleatoria. Para que esto se cumpla es necesario que su criterio de clasificación esté ligeramente relacionado con las características de las muestras a las que va a clasificar. Se puede decir que un clasificador débil tiene una tasa de error mayor que el 50%, por lo menos para las muestras de entrenamiento.

La mejor manera de explicar este algoritmo es paso a paso. En la Figura 4.1 se puede observar la generalización de Schapire y Singer [17] del algoritmo.

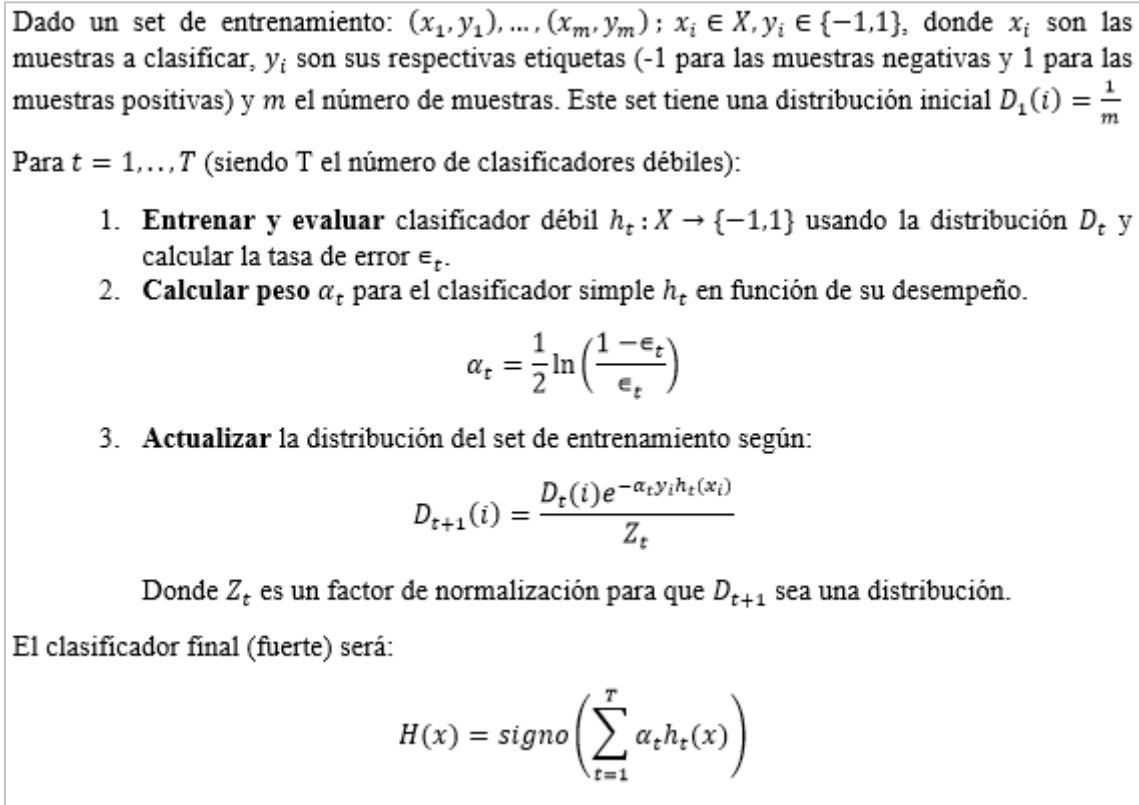


Figura 4.1. Algoritmo AdaBoost.

Primeramente, nótese que la distribución inicial de los datos es uniforme. Esto significa que al principio del algoritmo todas las muestras del set de entrenamiento tienen la misma importancia. Nótese también que en cada iteración se le asigna un peso α_t a un clasificador débil h_t .

En segundo lugar, obsérvese que $\alpha_t \geq 0$ si $\epsilon_t \leq \frac{1}{2}$ (lo cual se puede asumir por la definición de clasificador débil), y que α_t es mayor mientras más pequeña es ϵ_t . Esto significa que mientras menos error cometa el clasificador h_t , su decisión será más considerada que la de otros clasificadores.

Para explicar la actualización de la distribución que será utilizada en la siguiente iteración t , es necesario fijarse en el numerador de la ecuación, puesto que allí es donde en realidad se lleva a cabo la actualización de los pesos de cada muestra en la distribución D_{t+1} . En este documento se le llamará a dicho numerador $D_{act}(i)$, para cada muestra i .

$$D_{act}(i) = D_t(i) e^{-\alpha_t y_i h_t(x_i)} \quad (4.1)$$

Si $y_i = h_t(x_i)$ (el clasificador débil acierta), el producto $y_i h_t(x_i)$ vale 1, y en el caso contrario vale -1. Por lo tanto, para aquellas muestras que sean clasificadas correctamente por h_t , la exponencial se hace negativa, disminuyendo el peso de la muestra

en la distribución en mayor medida cuanto mayor sea α_t . En otras palabras, cuando una muestra es clasificada correctamente por un clasificador débil h_t , el siguiente clasificador h_{t+1} la considerará en menor medida cuanto mejor se haya desempeñado h_t . Por otro lado, para aquellas muestras i que sean clasificadas erróneamente por h_t , la exponencial se hace positiva, aumentando el peso de la muestra en mayor medida cuanto mayor sea α_t . Dicho de otra forma, cuando una muestra es clasificada equivocadamente por un clasificador débil h_t , el siguiente clasificador h_{t+1} la tomará más en cuenta cuanto mejor haya sido el desempeño de h_t .

Esto se debe a que si un clasificador débil es muy bueno (tiene un tasa de error baja), su α_t será mayor, y las muestras que clasifique correctamente no tendrán tanta importancia para los siguientes clasificadores débiles, mientras que las muestras que clasifique erróneamente deberían tener más importancia para los siguientes clasificadores por el hecho de que un clasificador anterior con buena ponderación (α_t) se equivocó al clasificarla.

El factor de normalización Z_t será igual al sumatorio de todos los pesos actualizados $D_{act}(i)$ de cada muestra para que D_{t+1} sea una distribución.

$$\begin{aligned} \sum_{i=1}^m D_{t+1}(i) = 1 &\rightarrow Z_t = \sum_{i=1}^m D_{act}(i) \\ &= \sum_{i=1}^m D_{act}(i) e^{-\alpha_t y_i h_t(x_i)} \end{aligned} \quad (4.2)$$

Una explicación muy aclaradora para este algoritmo puede observarse en la Figura 4.2.

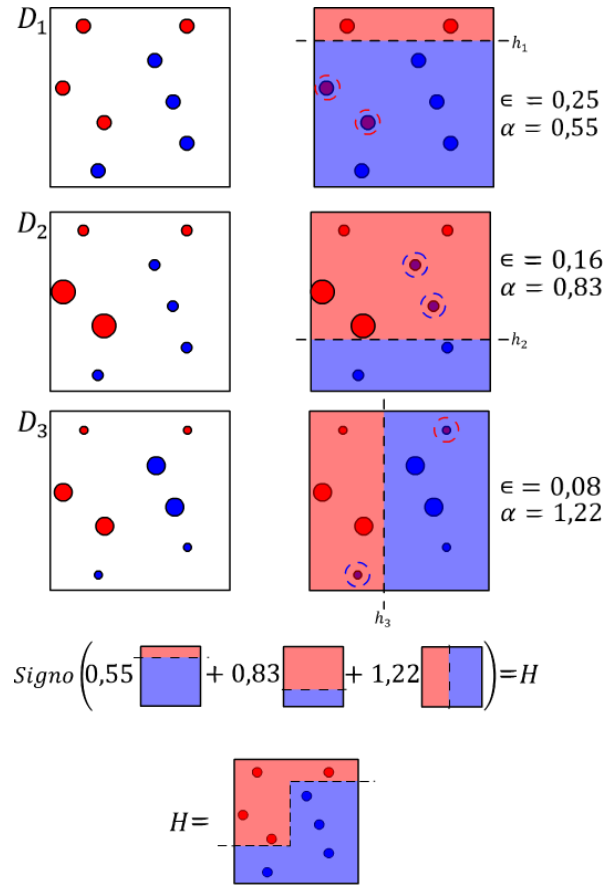


Figura 4.2. Demostración gráfica del algoritmo AdaBoost.

4.2. Detección facial usando el algoritmo de Viola-Jones

Como ya se ha comentado a lo largo del capítulo de detección, el algoritmo de detección facial de Viola-Jones utiliza una variante del AdaBoost, sin embargo, esto no es lo único que hace que la detección sea tan rápida y eficiente al mismo tiempo. A continuación se va a explicar paso a paso el algoritmo de Viola-Jones [14], junto con los conceptos necesarios para llevar a cabo cada uno de ellos.

4.2.1. Características

Las características que se usan para clasificar una imagen son unas características que hacen alusión a las *wavelets* de Haar, mas no lo son (ver capítulo 7.1. El análisis multirresolución y las *wavelets* en general). Por lo tanto, a lo largo de este documento se referirá a ellas como **características de Haar** (en inglés se utiliza el término *Haar-like features*, cuya traducción literal al español sería: características estilo Haar). Las características de Haar utilizadas por Viola-Jones tienen un aspecto como el de la Figura 4.3.

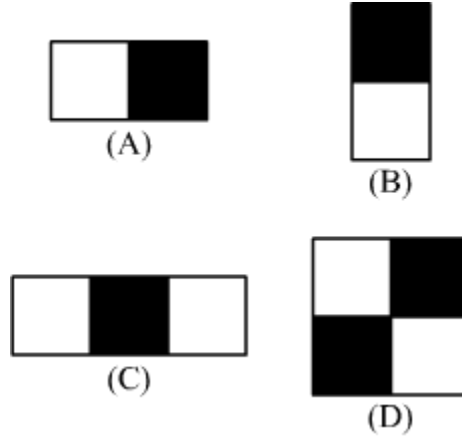


Figura 4.3. Tipos de características de Haar usadas por Viola-Jones.

Viola-Jones utilizaron originalmente esos tipos de características de Haar para su implementación, sin embargo, el número de características utilizadas varía dependiendo de la implementación [18] [19].

Cada característica es superpuesta sobre una imagen en todas las posiciones y a todos los tamaños posibles y el valor de la característica consiste en restar a la suma de todos los píxeles bajo zona negra de la característica, la suma de todos los píxeles bajo la zona blanca de la característica. Por ejemplo, en la Figura 4.4 se observa el cálculo de tres de las características de tipo (A) de la Figura 4.3 para una matriz de 2×4 .

$$\begin{array}{l}
 \begin{array}{cccc} 0 & 4 & 4 & 6 \\ 1 & 2 & 5 & 2 \end{array} \rightarrow \begin{array}{cccc} 0 & \boxed{4} & \boxed{4} & 6 \\ 1 & 2 & 5 & 2 \end{array} = (4)-(4) = 0 \\
 \\
 \begin{array}{cccc} 0 & 4 & 4 & 6 \\ 1 & 2 & 5 & 2 \end{array} \rightarrow \begin{array}{cccc} 0 & 4 & \boxed{4} & \boxed{6} \\ 1 & 2 & \boxed{5} & \boxed{2} \end{array} = (6+2)-(4+5) = -1 \\
 \\
 \begin{array}{cccc} 0 & 4 & 4 & 6 \\ 1 & 2 & 5 & 2 \end{array} \rightarrow \begin{array}{cccc} \boxed{0} & \boxed{4} & \boxed{4} & \boxed{6} \\ \boxed{1} & \boxed{2} & \boxed{5} & \boxed{2} \end{array} = (4+6+5+2)-(0+4+1+2) = 10
 \end{array}$$

Figura 4.4. Cálculo de tres de las característica de una matriz.

Una aportación de Lienhart [18] muy útil para el cálculo del número de características que puede haber en una imagen o porción de imagen de $W \times H$ (ancho por alto) es la siguiente fórmula:

$$K = XY \left(W + 1 - w \frac{X+1}{2} \right) \left(H + 1 - h \frac{Y+1}{2} \right) \quad (4.3)$$

$$X = \frac{W}{w}; \quad Y = \frac{H}{h}$$

Donde w y h son respectivamente el ancho y el alto mínimo del tipo de característica, y X e Y son los factores máximos de escalado de dicho tipo de característica.

Con lo cual el número exacto de la características de tipo (A) de la Figura 4.3 ($w = 2, h = 1$) que puede haber en una imagen de 24×24 sería 43200 características, y para los cuatro tipos de características de la misma figura sería 134.736 características.

4.2.2. Imagen integral

La suma de los píxeles de una imagen que se encuentran por debajo de una zona blanca o una zona negra de una característica de Haar, puede ser calculada rápidamente utilizando una representación intermedia de la imagen a la que Viola-Jones llaman **imagen integral**. El valor de una imagen integral para la una posición (x, y) es el resultado de la suma de todos los píxeles por arriba y hacia la derecha de esa posición, incluyendo el píxel de la posición indicada.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (4.4)$$

Donde $i(x, y)$ es la imagen de la que se desea calcular su imagen integral $ii(x, y)$. Por ejemplo, el cálculo de una imagen integral para una matriz de 2×3 sería:

$$\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \rightarrow \begin{pmatrix} a & a+b & a+b+c \\ a+d & a+b+d+e & a+b+c+d+e+f \end{pmatrix}$$

Obsérvese la Figura 4.5:

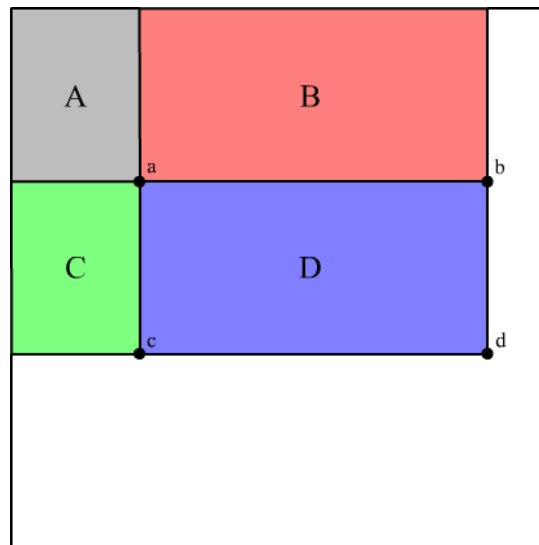


Figura 4.5. Zonas de una imagen cubiertas por los valores de su imagen integral en las posiciones a, b, c y d .

Es evidente que para calcular la suma rectangular de cualquier zona de una imagen, basta con acudir a cuatro posiciones en su correspondiente imagen integral. Nótese que el valor de una imagen integral en la posición a de la Figura 4.5 es el correspondiente al de la suma de todos los valores de la zona A en su respectiva imagen. El valor de la posición b es la de A+B, el de la posición c es la de A+C, y el de la posición d es la de A+B+C+D.

La suma de todos los valores de la zona D se calcula así: $d - b - c + a = d + a - (b + c)$.

La ventaja que presentan la utilización de imágenes integrales es formidable, ya que gracias a éstas el cálculo de las características se hace invariante al número de escalas y posiciones posibles de los distintos tipos de características de Haar disponibles, variando de forma muy despreciable el tiempo de ejecución al calcular números altos de características sobre una imagen o porción de imagen e igualando el tiempo de ejecución del cálculo de características del mismo tipo.

4.2.3. Clasificadores débiles y variante del algoritmo *Adaboost*

La filosofía que utilizan Viola-Jones para seleccionar los clasificadores débiles que se utilizarán para generar un clasificador fuerte es seleccionar de entre todas las características de Haar disponibles aquellas que mejor consigan separar a las muestras positivas (caras) de las negativas (no-caras).

Para cada característica de Haar se define un clasificador débil tal que el menor número de muestras sea clasificado de forma incorrecta. Por lo tanto, un clasificador débil $h_j(x)$ está formado por una característica f_j , un umbral θ_j y una paridad p_j . La polaridad indica la dirección de la desigualdad en la ecuación:

$$h_j(x) = \begin{cases} 1, & \text{Si } p_j f_j < p_j \theta_j \\ 0, & \text{En caso contrario} \end{cases} \quad (4.5)$$

Siendo x una imagen o porción de imagen de $W \times H$. Viola-Jones utilizaron para su implementación una **resolución base** de 24×24 píxeles.

Viola-Jones afirman que se puede utilizar cualquier estrategia de aprendizaje automático partiendo de los clasificadores simples propuestos. No obstante, optan por utilizar una variante del AdaBoost de Freund y Schapire [16]. El algoritmo final puede apreciarse en la Figura 4.6.

- Dadas las imágenes de muestra $(x_1, y_1), \dots, (x_n, y_n)$ donde $y_i = 0, 1$ para muestras negativas y positivas, respectivamente.
- Inicializar los pesos $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ para $y_i = 0, 1$ respectivamente, donde m y l son el número de muestras negativas y positivas, respectivamente.
- Para $t = 1, \dots, T$:

1. Normalizar los pesos,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

De tal manera que w_t sea una distribución de probabilidad.

2. Para cada característica j , entrenar un clasificador h_j que esté restringido a utilizar únicamente una característica. El error es evaluado con respecto a w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
3. Elegir el clasificador h_t , que tenga el menor error ϵ_t .
4. Actualizar los pesos:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

Donde $e_i = 0$ Si la muestra x_i es clasificada correctamente, $e_i = 1$ en caso contrario, y $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- El clasificador fuerte sería:

$$h(x) = f(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{En caso contrario} \end{cases}$$

Donde $\alpha_t = \log \frac{1}{\beta_t}$

Figura 4.6. AdaBoost modificado de Viola-Jones.

Nótese que a pesar de ser $w_{t,i}$ una distribución, las muestras negativas se tratan por separado de las muestras positivas, de tal manera que inicialmente podrían utilizarse un número distinto de muestras positivas y negativas en el set de entrenamiento.

Obsérvese ahora la fórmula de actualización de los pesos:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \quad (4.6)$$

$$\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$$

Nótese que $\beta_t \leq 1$ para valores de $\epsilon_t \geq \frac{1}{2}$, disminuyendo su valor en mayor medida cuanto menor sea el error, mientras que para valores de $\epsilon_t < \frac{1}{2}$, los valores de β aumentan por encima de uno cuanto mayor sea ϵ_t . Se indica además que cuando una muestra es clasificada correctamente, $e_i = 0$, y en consecuencia, si $\epsilon_t > \frac{1}{2}$, el peso de dicha muestra disminuye más cuanto menor haya sido el error de las muestras, y si $\epsilon_t < \frac{1}{2}$, el peso de dicha muestra aumenta más cuanto peor haya sido el clasificador (mayor ϵ_t). Como en cada

iteración se elige al mejor clasificador, el caso de que éste tenga un $\epsilon_t > \frac{1}{2}$ no evita que aun así haya sido el mejor clasificador, por lo tanto, el peso de la muestra aumenta para que en la siguiente iteración el próximo clasificador se esfuerce más sobre ella. De igual manera es muy extraño que ϵ_t tome valores excesivamente altos. Para el caso en que el clasificador débil se haya equivocado al clasificar la muestra, $e_t = 1$, igualado a cero la potencia de β_t y manteniendo el peso de la muestra.

Es importante tomar en cuenta que al ser una adaptación del algoritmo para un entorno como el que se presenta, es posible muy probable que $\epsilon_t > \frac{1}{2}$ en más de una ocasión. Esto se debe al número de características limitado (a pesar de ser muy alto), y que al ser éstas tan simples, su comportamiento a partir de cierto número de iteraciones puede llegar a ser más inesperado.

El hecho de que se le asigne mayor o menor importancia a una muestra clasificada correctamente que a una clasificada incorrectamente (los pesos de éstas últimas se mantienen) es una de las peculiaridades de esta variante.

Lo siguiente es observar a la ecuación que describe al clasificador fuerte, donde hay tres detalles que llaman la atención:

$$h(x) = f(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{En caso contrario} \end{cases} \quad (4.7)$$

$$\alpha_t = \log \frac{1}{\beta_t}$$

El primero de ellos es que el valor de α_t es mayor que cero para valores de β_t menores que uno (el mejor clasificador débil de la iteración t es bueno), y en el caso contrario, α_t es mayor que cero para valores de β_t mayores que uno (el mejor clasificador débil de la iteración t es malo).

El segundo es que al ser las etiquetas 0, para muestras negativas o 1, para muestras positivas, a la hora de utilizar el clasificador fuerte h , la decisión de aquellos clasificadores débiles que no consideran que la entrada x sea un rostro, no será tomada en cuenta; sin embargo, el peso α_t que tiene dicho clasificador sí que influye en el umbral de decisión (parte derecha de la desigualdad). De hecho, la desigualdad que aparece en la ecuación se puede interpretar de la siguiente manera:

Cuando una porción de imagen x entra al clasificador fuerte h , se decide que es un rostro si la suma de todas las ponderaciones α_t de los clasificadores débiles h_t que así lo consideran, supera cierto umbral dependiente de la suma total de las ponderaciones α_t de todos los clasificadores débiles que forman al clasificador fuerte.

O lo que es lo mismo, si se escribe la desigualdad de la siguiente manera, se puede tomar a la parte izquierda de la desigualdad como una distribución de probabilidad:

$$\frac{\sum_{t=1}^T \alpha_t h_t(x)}{\sum_{t=1}^T \alpha_t} \geq \frac{1}{2} \quad (4.8)$$

Cuando una porción de imagen x entra al clasificador fuerte h , se decide que es un rostro si la decisión conjunta de ponderada de todos los clasificadores débiles h_t que lo forman es mayor que cierto porcentaje.

El tercer detalle que llama la atención es el $\frac{1}{2}$ de la desigualdad. Nótese que en las interpretaciones del autor de este documento se ignora el hecho de la existencia de dicho valor diciendo “cierto umbral dependiente de la suma total de las ponderaciones α_t ”, en lugar de “la mitad de la suma total de las ponderaciones α_t ” en la primera interpretación o “cierto porcentaje” en lugar de “el 50%”, en la segunda interpretación.

Esto se debe a que existen dos tipos de clasificaciones erróneas: falsos positivos (una no-cara clasificada como tal) y falsos negativos (una cara erróneamente clasificada). En objetivo principal de un clasificador fuerte en el algoritmo de Viola-Jones (luego se volverá a remarcar cuando se expliquen los clasificadores en cascada) es mantener nula la tasa de falsos negativos (que equivale a una tasa de detección del 100% durante el entrenamiento) sin importar que esto implique que no se reduzca ampliamente la tasa de falsos positivos. Por lo tanto Viola-Jones indican que “generalmente un menor umbral conlleva a mayores tasas de detección y mayores tasas de falsos positivos”. Finalmente, como el sumatorio de todas las α_t es constante, entonces lo único que puede cambiar el umbral es el $\frac{1}{2}$.

Es curioso observar que los mejores clasificadores de las primeras iteraciones suelen ser tales que se hace literalmente visible la forma que tienen de describir un rostro. Por ejemplo, en la Figura 4.7 se muestra el resultado de los 2 mejores clasificadores simples que obtuvieron Viola y Jones en su implementación [14]. La primera de ellas indica que la región de los ojos suele ser más oscura que la de las mejillas, mientras que la segunda compara la luminancia de la región de los ojos con la del tabique nasal.

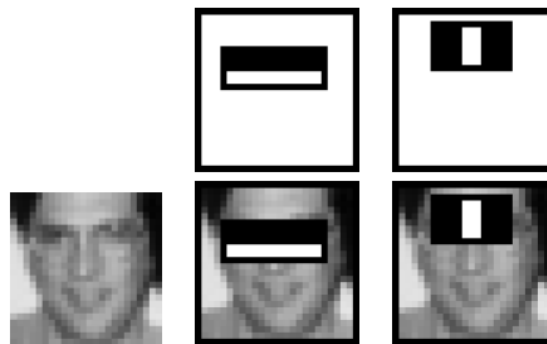


Figura 4.7. Los dos mejores clasificadores simples que obtuvieron Viola y Jones en su implementación.

4.2.4. Clasificadores en cascada

Para aumentar la velocidad de detección, Viola-Jones proponen entrenar cierto número de clasificadores fuertes para luego conectarlos en cascada tal y como se indica en la Figura 4.8.

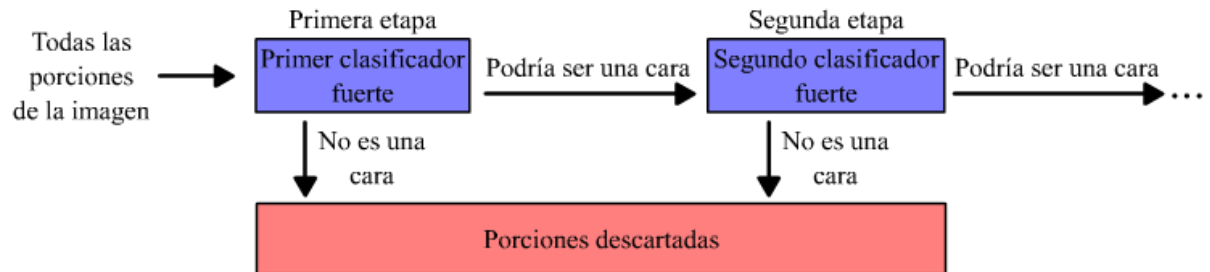


Figura 4.8. Clasificadores en cascada. Las porciones que superen la última etapa serán aquellas clasificadas como caras por el detector.

Cada clasificador fuerte está formado por un mayor número de clasificadores débiles cuanto mayor sea su posición en la cascada, dando a entender que a medida de que una porción de imagen **avanza** en la cascada, el siguiente clasificador será más estricto, pues al ser formado por más clasificadores débiles, debería ser más preciso. Como se puede observar también de la Figura 4.8, a los clasificadores fuertes de la cascada se les llama **etapa**.

Nótese que cada clasificador deja pasar una tasa de detección (idealmente un 100%, debiéndose cumplir esta tasa durante el entrenamiento) junto con una tasa de falsos positivos, mientras que aquellas porciones que son clasificadas como no-caras son descartadas permanentemente. Esto equivale a que durante el entrenamiento, cada clasificador debe mantener o disminuir el umbral para que la tasa de detección en la práctica sea lo mayor posible.

Viola-Jones utilizaron un detector formado por 38 etapas, entrenadas cada una con 4.916 muestras de caras más sus respectivas reflexiones verticales (9.832 en total) y 10.000 muestras negativas. Todas las imágenes de entrenamiento fueron de 24x24 píxeles. Las muestras negativas fueron porciones de imagen de 9.544 imágenes inspeccionadas manualmente para asegurarse de que no contienen ninguna cara. Dentro de esas 9.544 imágenes existen alrededor de 350 millones de porciones de imagen (muestras negativas). Para la primera etapa de la cascada utilizaron un conjunto aleatorio de entre todas las muestras negativas, mientras que para las siguientes etapas, utilizaron los falsos positivos de las etapas anteriores. El número de características (clasificadores débiles) de las primeras cinco etapas fue de 1, 10, 25, 25 y 50, mientras que para las demás etapas el número de características iba aumentando cada vez más, consiguiendo un detector de 6.061 características.

4.2.5. Otras consideraciones

Dependiendo del desplazamiento del barrido y el factor de escalado de las características de Haar, es posible que una cara sea detectada varias veces por el detector, por lo que si dos sub-ventanas (porciones de imagen) son detectadas como caras y se solapan, entonces el detector las integrará como una sola.

Una última consideración es que tanto todas las imágenes de entrenamiento, como las porciones de la imagen de entrada al detector, fueron pre-procesadas antes de entrar al detector, normalizando su varianza para minimizar el efecto de las condiciones de iluminación. Viola-Jones indican que este pre-procesado es coherente ya que puede llevarse a cabo utilizando imágenes integrales.

5. Análisis por componentes principales (PCA)

El análisis por componentes principales (PCA, por sus siglas en inglés) es una poderosa técnica estadística que tiene como objetivo principal reducir la dimensión de un conjunto de muestras para representarlas en un espacio donde se pueda interpretarlas de mejor manera. Esto se consigue proyectando los datos sobre un espacio dentro del cual no exista teóricamente covarianza entre las variables que forman a cada vector de observación.

5.1. Conceptos básicos y nomenclatura

El objetivo de este apartado es principalmente aclarar nomenclatura y adicionalmente hacer un breve repaso de algunos conceptos básicos.

Para empezar, es importante comentar que a lo largo de este documento los vectores serán representados en negrita.

Se denomina **vector de observación** al vector columna cuyas filas representan cada variable de una muestra contenida en una colección, o población, de datos multivariados. A menudo se utilizarán indistintamente los términos vector de observación y muestra, debido a que el vector de observación es el que se utiliza para representar las características (o variables aleatorias) de una muestra

$$\mathbf{x}_{nx1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{nx1} \quad (5.1)$$

Donde n es la dimensión de los datos multivariados o número de variables que representan cada muestra.

Una **matriz de observaciones**, por lo tanto, se refiere a una matriz cuyas columnas son los vectores de observación de un conjunto de muestras, y en consecuencia las filas son las características.

$$X_{nxM} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,M} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,M} \end{bmatrix}_{nxM} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_M]_{nxM} \quad (5.2)$$

Donde el primer subíndice de $x_{j,k}$ representa el número de característica y el segundo el número de muestra, y M es el número de muestras.

Sea una matriz de observaciones X_{nxM} , la **media** se conoce como el promedio o la tendencia central que tienen cada uno de los vectores que forman a X .

$$\boldsymbol{\mu}_{nxM} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}_{nxM} = \frac{1}{M} \sum_{k=1}^M \mathbf{x}_k \quad (5.3)$$

Se llama $\hat{\mathbf{x}}$ al resultado de restar a un vector de observaciones la media.

$$\hat{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu} \quad (5.4)$$

Dicho esto, cuando a todas las columnas de una matriz de observaciones se le resta la media, a la matriz resultante se dice que está en **forma de desviación media**.

$$A_{n \times M} = [\hat{\mathbf{x}}_1 \quad \hat{\mathbf{x}}_2 \quad \cdots \quad \hat{\mathbf{x}}_M]_{n \times M} = \begin{bmatrix} \hat{x}_{1,1} & \hat{x}_{1,2} & \cdots & \hat{x}_{1,M} \\ \hat{x}_{2,1} & \hat{x}_{2,2} & \cdots & \hat{x}_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{n,1} & \hat{x}_{n,2} & \cdots & \hat{x}_{n,M} \end{bmatrix}_{n \times M} \quad (5.5)$$

$$\rightarrow A_{n \times M} = \begin{bmatrix} x_{1,1} - \mu_1 & x_{1,2} - \mu_1 & \cdots & x_{1,M} - \mu_1 \\ x_{2,1} - \mu_2 & x_{2,2} - \mu_2 & \cdots & x_{2,M} - \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} - \mu_n & x_{n,2} - \mu_n & \cdots & x_{n,M} - \mu_n \end{bmatrix}_{n \times M}$$

Para una determinada variable aleatoria o característica, la **varianza** se define como una medida de dispersión y se calcula de la siguiente manera:

$$Var(x_j) = \frac{1}{M} \sum_{k=1}^M (x_{j,k} - \mu_j)^2 \quad (5.6)$$

Una excelente interpretación de la varianza, observando la fórmula, sería la distancia media de las muestras de una población a la media.

Análogamente se define la **covarianza**, como el grado de variación conjunta que existe entre dos variables aleatorias.

$$Cov(x_j, x_i) = \frac{1}{M} \sum_{k=1}^M (x_{j,k} - \mu_j)(x_{i,k} - \mu_i) \quad (5.7)$$

Finalmente, se define la **matriz de covarianzas** como la matriz simétrica cuya diagonal está formada por las varianzas de cada una de las variables aleatorias, y el resto de valores las covarianzas entre ellas. La matriz de covarianzas de una matriz de observación en forma de desviación media se forma con la suma de sus columnas multiplicada por su transpuesta, o lo que es lo mismo, multiplicando la matriz por su transpuesta.

$$S_x = \frac{1}{M} \sum_{k=1}^M (\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T) = \frac{1}{M} A A^T \quad (5.8)$$

$$\rightarrow S_x = \begin{bmatrix} (\hat{x}_{1,1})^2 + (\hat{x}_{1,2})^2 + \dots + (\hat{x}_{1,M})^2 & (\hat{x}_{1,1}\hat{x}_{2,1}) + (\hat{x}_{1,2}\hat{x}_{2,2}) + \dots + (\hat{x}_{1,M}\hat{x}_{2,M}) & \dots & (\hat{x}_{1,1}\hat{x}_{n,1}) + (\hat{x}_{1,2}\hat{x}_{n,2}) + \dots + (\hat{x}_{1,M}\hat{x}_{n,M}) \\ (\hat{x}_{1,1}\hat{x}_{2,1}) + (\hat{x}_{1,2}\hat{x}_{2,2}) + \dots + (\hat{x}_{1,M}\hat{x}_{2,M}) & (\hat{x}_{2,1})^2 + (\hat{x}_{2,2})^2 + \dots + (\hat{x}_{2,M})^2 & \dots & (\hat{x}_{2,1}\hat{x}_{n,1}) + (\hat{x}_{2,2}\hat{x}_{n,2}) + \dots + (\hat{x}_{2,M}\hat{x}_{n,M}) \\ \vdots & \vdots & \ddots & \vdots \\ (\hat{x}_{1,1}\hat{x}_{n,1}) + (\hat{x}_{1,2}\hat{x}_{n,2}) + \dots + (\hat{x}_{1,M}\hat{x}_{n,M}) & (\hat{x}_{2,1}\hat{x}_{n,1}) + (\hat{x}_{2,2}\hat{x}_{n,2}) + \dots + (\hat{x}_{2,M}\hat{x}_{n,M}) & \dots & (\hat{x}_{n,1})^2 + (\hat{x}_{n,2})^2 + \dots + (\hat{x}_{n,M})^2 \end{bmatrix}_{n \times n}$$

$$\rightarrow S_x = \begin{bmatrix} Var(x_1) & Cov(x_1, x_2) & \dots & Cov(x_1, x_n) \\ Cov(x_1, x_2) & Var(x_2) & \dots & Cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(x_1, x_n) & Cov(x_2, x_n) & \dots & Var(x_n) \end{bmatrix}_{n \times n} \quad (5.9)$$

Donde cada subíndice indica la variable aleatoria que representa. Obsérvese que la diagonal de S_x contiene todas las varianzas de cada una de las variables aleatorias consideradas, y por tanto la **traza** (suma de los valores de la diagonal) de la matriz de covarianzas representa la varianza total de los datos.

$$tr(S_x) = \frac{1}{M} [Var(x_1) + Var(x_2) + \dots + Var(x_n)] \quad (5.10)$$

5.2. El análisis por componentes principales

Desde el punto de vista algebraico, el análisis por componentes principales equivale a un cambio de variable del tipo:

$$X = PY \quad (5.11)$$

Donde X es la matriz de observaciones en forma de desviación media, formada por los vectores de observación $[\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_M]$; P es una matriz ortonormal; Y son las coordenadas de cada vector de X sobre el subespacio que forman las columnas de P y M es el número de muestras de X , cada una con n características.

Dicho esto, se procede a calcular el valor de las columnas de P . Partiendo de la fórmula (5.11), se despeja Y :

$$Y = P^{-1}X \quad (5.12)$$

Donde se sabe que $P^{-1} = P^T$, debido a que las columnas de P son ortonormales. En consecuencia, la matriz de covarianzas de Y quedaría:

$$S_y = YY^T = (P^T X)(P^T X)^T = P^T XX^T P = P^T S_x P \quad (5.13)$$

Por lo que recordando la forma de la matriz de covarianzas (5.9), y que el objetivo era que no existiese covarianza entre las coordenadas de los vectores de observación en el nuevo espacio, entonces S_y no es más que una matriz diagonal. Despejando se obtiene:

$$S_x = PS_y P^T = PS_y P^{-1} \quad (5.14)$$

Llegando a la conclusión de que S_y sería el resultado de la diagonalización de la matriz de covarianzas de S_x .

Es evidente que la matriz de covarianzas S_x es simétrica, y por lo tanto diagonalizable ortogonalmente (Teorema espectral para matrices simétricas [20]). Además, según el teorema de la diagonalización, las columnas de P son los vectores propios $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ de S_x , correspondientes a cada valor propio $\lambda_1, \lambda_2, \dots, \lambda_n$, que a su vez estarán ubicados en la diagonal de la matriz S_y . Dichos vectores propios son también llamados **componentes principales**, y para hacer de P una matriz ortonormal, se divide cada uno de ellos entre su norma.

Resumiendo: Sean $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ los vectores propios de la matriz S_x , y $\lambda_1, \lambda_2, \dots, \lambda_n$ sus correspondientes autovalores, entonces:

$$[S_x]_{n \times n} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_n]_{n \times n} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}_{n \times n} \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix}_{n \times n} \quad (5.15)$$

$$= P S_y P^T$$

Volviendo a la estructura de la matriz de covarianzas (5.9), cada valor de la diagonal de S_y (los autovalores calculados) representa la varianza de los datos de Y a lo largo de su respectivo autovector, por lo que se puede permitir una reducción de dimensión de los datos contenidos en X proyectándolos sobre un nuevo subespacio formado por los autovectores resultantes cuyos valores propios sean los más altos sin perder apenas información. De hecho, como es sabido, en un cambio ortogonal de variables $X = PY$ no cambia la varianza total de los datos [20], y por tanto la varianza total de X (traza de S_x) es igual a la varianza total de Y (traza de S_y). Dicho esto último, puede medirse la fracción de varianza total (información) que representa cada autovector como:

$$\%Var_{Total} = \frac{Var(y_k)}{Var_{Total}} 100 = \frac{Var_{componente\ principal\ k}}{Var_{Total}} 100 \quad (5.16)$$

$$= \frac{\lambda_k}{Var_{Total}} 100$$

Finalmente, y como ya se mencionó en el párrafo anterior, si se toman los L autovectores correspondientes a los L autovalores más altos, entonces las dimensiones de P serán $n \times L$ y las coordenadas de los vectores de observación de la matriz X se calcularían de la siguiente forma:

$$\mathbf{y}_{L \times 1} = [P]_{L \times n}^T \mathbf{x}_{n \times 1} \quad (5.17)$$

En la Figura 5.1 se puede ver una representación gráfica de lo explicado. Se trata de un conjunto de 200 muestras generadas aleatoriamente en MATLAB de manera que

sigan una distribución normal. En la Figura 5.1 (a) se puede ver el diagrama de dispersión de los datos, en la Figura 5.1 (b) se observan los mismos datos en forma de desviación media, y en la Figura 5.1 (c) se ven representados como flechas de color la primera componente principal (rojo) y la segunda componente principal (verde).

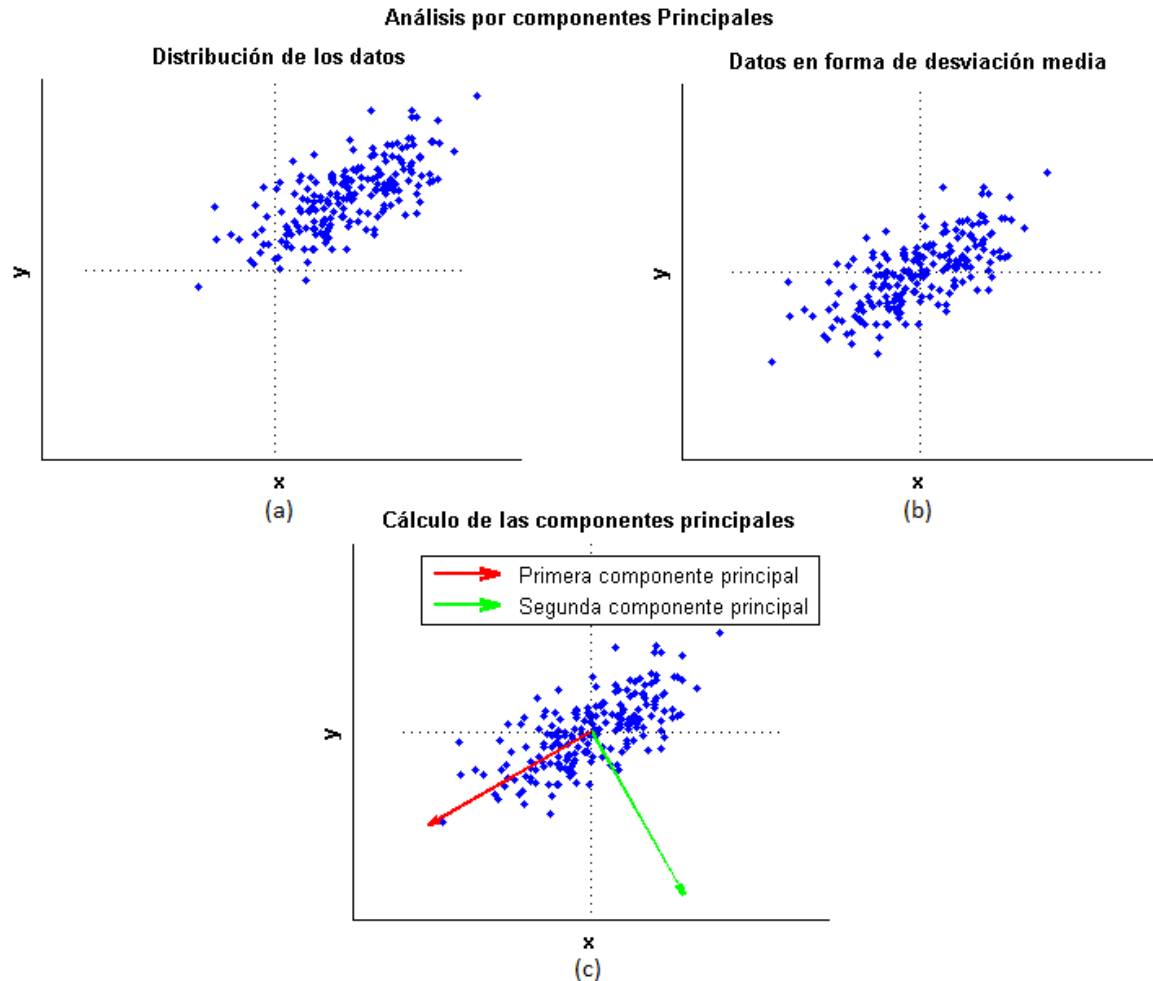
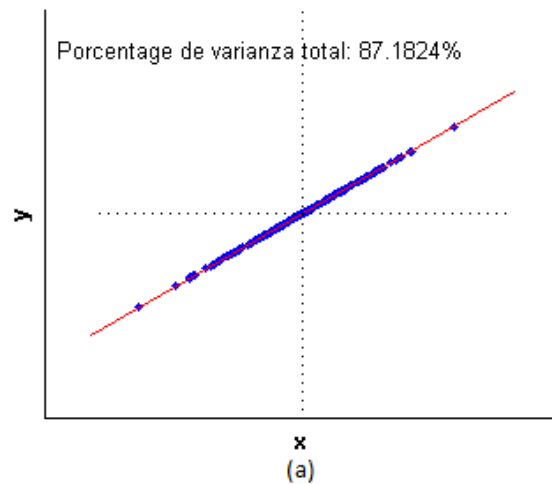


Figura 5.1. Procedimiento de cálculo de las componentes principales. Las gráficas fueron obtenidas generando una distribución aleatoria en MATLAB y calculando sus componentes principales.

En la Figura 5.2 se pueden observar las proyecciones de los datos de la Figura 5.1 sobre cada una de sus componentes principales. De esta manera se comprueba que la pérdida de información (varianza) es menor al reducir la dimensión de los datos proyectándolos sobre la primera componente principal (Figura 5.2.a) que sobre la segunda (Figura 5.2.b).

Proyección de los datos sobre la primera componente principal



Proyección de los datos sobre la segunda componente principal

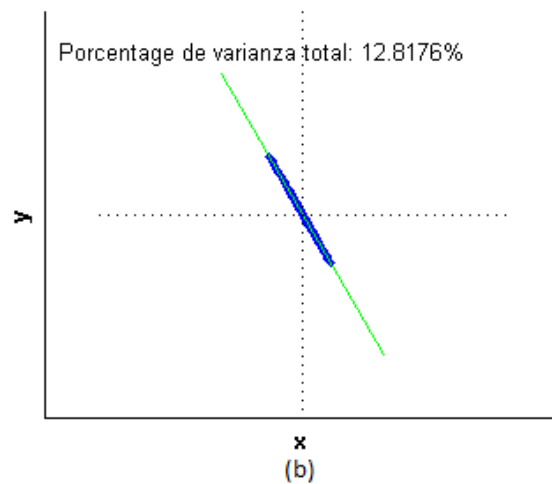


Figura 5.2. Proyección de los datos de la Figura 5.1 sobre cada una de las componentes principales.

5.3. Reconocimiento facial utilizando *eigenfaces*.

Desarrollado por Turk y Pentland [9], este tipo de reconocimiento facial se basa en el análisis por componentes principales para identificar patrones en las imágenes y así clasificar a cada individuo en función de sus coordenadas o pesos del espacio formado por las componentes principales.

Para llevar a cabo cualquier tarea de clasificación es necesario un set de entrenamiento, por lo tanto se toma una colección de imágenes de rostros que harán el papel de vectores de observación, cuyas filas (características) están formadas por cada píxel de la imagen. En consecuencia, todas las imágenes deben tener el mismo tamaño con número de píxeles $n = W \times H$, siendo W la anchura de las imágenes y H la altura. Esto significa que cada imagen es representada como un punto en un espacio vectorial de n dimensiones. Se define **vector imagen** como el vector de observación correspondiente a una imagen (Figura 5.3).

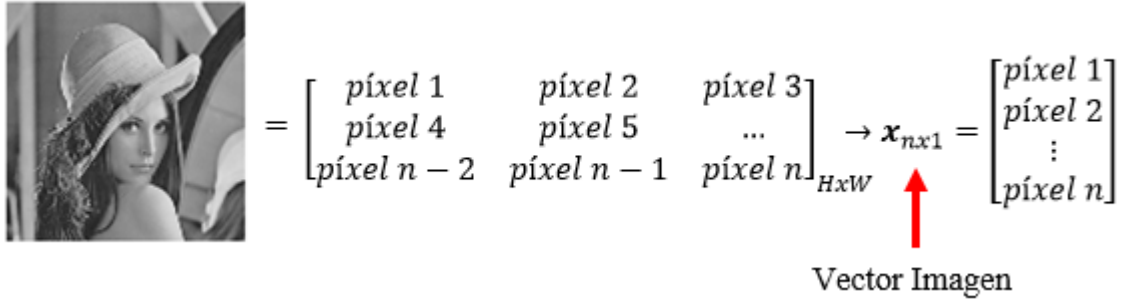


Figura 5.3. Formación de un vector imagen.

Por ejemplo, Turk y Pentland desarrollaron su investigación utilizando imágenes 256x256 píxeles, con lo que las imágenes estaban contenidas en un espacio vectorial de 65536 dimensiones. Por consiguiente, es evidente que es necesaria una reducción de dimensión a un espacio vectorial eligiendo las características que mejor representan los diferentes patrones. Parece irrefutable la decisión de utilizar el PCA para este tipo de tarea.

Si se llama $X_{n \times M}$ a la matriz de observaciones, siendo M el número de imágenes de entrenamiento y sus columnas son los vectores imagen de cada una de las imágenes de entrenamiento, y se llama $A_{n \times M}$ a la forma de desviación media de dicha matriz, entonces su matriz de covarianzas sería:

$$S_x = \frac{1}{M} \sum_{k=1}^M (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T = \frac{AA^T}{M} \quad (5.18)$$

El inconveniente que tiene este procedimiento es que la matriz S_x resultante es de $n \times n$ y para vectores imagen faciales típicos, n es un valor del orden de las decenas de miles, y en consecuencia el cálculo de los autovectores resulta irrealizable. Por suerte, las dimensiones de la matriz $A^T A$ son más adecuadas ($M \times M$), puesto que la cantidad de imágenes de entrenamiento suele ser mucho menor que las dimensiones de los vectores imagen, pero ¿cómo se relacionan los autovectores y autovalores de la matriz $A^T A$ con los de la matriz S_x ?

Los autovectores \mathbf{u}_i de, $A^T A$ por definición cumplen que:

$$\frac{1}{M} A^T A \mathbf{u}_i = \sigma_i \mathbf{u}_i \quad (5.19)$$

Siendo σ_i el autovalor correspondiente a dicho autovector.

Se multiplica por la izquierda a ambos miembros por la matriz A resultando:

$$\frac{1}{M} AA^T A \mathbf{u}_i = S_x(A \mathbf{u}_i) = \sigma_i(A \mathbf{u}_i) \quad (5.20)$$

También, por definición, los autovectores y autovalores de S_x cumplen que:

$$[S_x]_{n \times n} [\mathbf{v}_i]_{n \times 1} = \lambda_i [\mathbf{v}_i]_{n \times 1} \quad (5.21)$$

Por tanto, es elemental que los vectores $A_{n \times M} [\mathbf{u}_i]_{M \times 1}$ tienen los mismos $M - 1$ autovalores que los vectores propios de S_x correspondientes a los $M - 1$ mayores autovalores de S_x (el M autovalor es prácticamente nulo), debido a que ambos grupos de vectores cumplen la condición de autovector para S_x , como se puede observar en (5.20) y (5.21). Además, los vectores $A \mathbf{u}_i$ son iguales o son múltiplos de \mathbf{v}_i . Luego, el conjunto de autovectores que forman la matriz P será:

$$P_{n \times M} = A_{n \times M} [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_M] \quad (5.22)$$

Pero Turk y Pentland no hablan de la varianza que tienen los datos en el nuevo espacio calculado con la matriz $A^T A$. A primera vista, parece que como el resultado final posee menos autovectores, se ha perdido información (varianza), lo cual contradiría lo dicho sobre los cambios ortogonales de variables, por lo que a continuación se presenta un procedimiento que demuestra que sí se mantiene la varianza total de los datos. En el capítulo de implementación se saca provecho de esta información.

Para ello se observa primero la forma de la matriz $A^T A$:

$$\frac{1}{M} A^T A = \frac{1}{M} \begin{bmatrix} \hat{\mathbf{x}}_1^T \\ \hat{\mathbf{x}}_2^T \\ \vdots \\ \hat{\mathbf{x}}_n^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_1 & \hat{\mathbf{x}}_2 & \cdots & \hat{\mathbf{x}}_n \end{bmatrix} \quad (5.23)$$

No obstante, para esta demostración sólo hace falta saber los valores de su diagonal:

$$\begin{aligned} \frac{1}{M} A^T A &= \frac{1}{M} \begin{bmatrix} (x_{1,1} - \mu_1)^2 + (x_{2,1} - \mu_2)^2 + \cdots + (x_{n,1} - \mu_n)^2 & \cdots & \cdots & \cdots \\ \vdots & (x_{1,2} - \mu_1)^2 + (x_{2,2} - \mu_2)^2 + \cdots + (x_{n,2} - \mu_n)^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \cdots & \cdots & (x_{1,M} - \mu_1)^2 + (x_{2,M} - \mu_2)^2 + \cdots + (x_{n,M} - \mu_n)^2 \end{bmatrix}_{M \times M} \\ \rightarrow tr \left(\frac{1}{M} A^T A \right) &= \frac{1}{M} \left\{ \begin{aligned} &[(x_{1,1} - \mu_1)^2 + (x_{2,1} - \mu_2)^2 + \cdots + (x_{n,1} - \mu_n)^2] + \\ &[(x_{1,2} - \mu_1)^2 + (x_{2,2} - \mu_2)^2 + \cdots + (x_{n,2} - \mu_n)^2] + \cdots + \\ &[(x_{1,M} - \mu_1)^2 + (x_{2,M} - \mu_2)^2 + \cdots + (x_{n,M} - \mu_n)^2] \end{aligned} \right\} \\ \rightarrow tr \left(\frac{1}{M} A^T A \right) &= \frac{1}{M} \left\{ \begin{aligned} &[(x_{1,1} - \mu_1)^2 + (x_{1,2} - \mu_1)^2 + \cdots + (x_{1,M} - \mu_1)^2] + \\ &[(x_{2,1} - \mu_2)^2 + (x_{2,2} - \mu_2)^2 + \cdots + (x_{2,M} - \mu_2)^2] + \cdots + \\ &[(x_{n,1} - \mu_n)^2 + (x_{n,2} - \mu_n)^2 + \cdots + (x_{n,M} - \mu_n)^2] \end{aligned} \right\} \\ \rightarrow tr \left(\frac{1}{M} A^T A \right) &= \frac{1}{M} [Var(x_1) + Var(x_2) + \cdots + Var(x_n)] \quad (5.24) \\ &= tr(S_x) \end{aligned}$$

Así se demuestra que los autovalores de $A^T A$ son iguales a los mejores $M - 1$ autovalores de AA^T (porque la varianza total de Y es la misma), y como su suma representa la varianza total de los datos iniciales, en la práctica se puede comprobar que el resto de los autovalores de AA^T , si no son cero, son prácticamente nulos. En resumen, a pesar de que el número de autovalores y autovectores resultantes a partir de $A^T A$ es menor que a partir de AA^T , la varianza original de los datos se mantiene de igual forma.

Al conjunto de los autovectores resultantes se les llama *eigenfaces*, cuya traducción literal al español sería “autocaras”, o “caras propias”; sin embargo, al no existir ninguna traducción oficial, a lo largo de este documento serán llamadas *eigenfaces*. Este término tiene su origen en que si se vuelven a redimensionar las *eigenfaces* al tamaño de las imágenes de entrenamiento y se expanden para que por ejemplo, su valor mínimo sea cero y el máximo sea 255 (valores mínimo y máximo de un píxel para una imagen de 8 bits), al ser visualizadas como imágenes corrientes, pueden observarse caras.

En la Figura 5.4 se muestra un set de entrenamiento formado por las primeras 10 imágenes de cada individuo de la base de datos de *Yale University* (15 individuos) [21]. En las Figura 5.5 y Figura 5.6 se muestran respectivamente la imagen media y las primeras 20 *eigenfaces* de dicho set calculadas en MATLAB.

En la Figura 5.6 se indica además el porcentaje de varianza total acumulado que explica cada una de las 20 primeras *eigenfaces*. Por lo tanto, para saber el porcentaje de varianza explicado por una de las *eigenfaces* en concreto, habría que restar al porcentaje de varianza acumulado de dicha *eigenface* el de la anterior.



Figura 5.4. Set de entrenamiento formado por algunas de las imágenes de la base de datos de Yale University [21].



Figura 5.5. Imagen media del set de entrenamiento de la Figura 5.4. Esta figura fue obtenida utilizando MATLAB.



Figura 5.6. Veinte primeras *eigenfaces* del set de entrenamiento de la Figura 5.4. Esta figura fue obtenida utilizando MATLAB.

Una vez calculadas las *eigenfaces*, el siguiente paso es reducir la dimensión de los datos. Para ello se descartan las *eigenfaces* correspondientes a los autovalores más bajos y se utilizan solamente aquellas correspondientes a los valores más altos.

Para terminar con la etapa de entrenamiento, se guarda para cada uno de los individuos del conjunto de entrenamiento un patrón en función de la proyección de sus imágenes de entrenamiento sobre el subespacio formado por las *eigenfaces*, también llamado **espacio de caras**.

$$\mathbf{y} = \mathbf{P}^T \mathbf{x} \quad (5.25)$$

Cuando se inicia el sistema con el fin de llevar a cabo un reconocimiento, la imagen de entrada es proyectada en el espacio de caras (habiéndole restado previamente la media de entrenamiento). Posteriormente, se usan las coordenadas resultantes para tomar una

decisión basada en los patrones guardados durante el módulo de inscripción. Por ejemplo, una regla de decisión podría ser verificar si la distancia euclídea que hay desde de la proyección de dicha imagen de entrada hasta las coordenadas medias de las imágenes de entrenamiento de un individuo en el espacio de caras es lo suficientemente pequeña (se establece un umbral) para ser clasificada como una muestra de dicho individuo.

En resumen, para llevar a cabo el reconocimiento facial de un vector imagen de entrada \mathbf{x}' siguiendo la regla de la distancia euclídea, el procedimiento a seguir es el siguiente:

1. Normalizar la imagen de entrada $(\mathbf{x}' - \boldsymbol{\mu})$ y proyectar el resultado en el espacio de caras formado por las *eigenfaces* obtenidas en el entrenamiento:

$$\mathbf{y}' = P^T(\mathbf{x}' - \boldsymbol{\mu}) = \begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ y_m' \end{bmatrix} \quad (5.26)$$

Donde m representa el número de *eigenfaces* utilizadas, e \mathbf{y}' el patrón de la imagen de entrada.

2. Calcular la distancia euclídea d entre el patrón de entrada y el patrón almacenado $\boldsymbol{\Omega}_I$ para el individuo a verificar.

$$d = \|\boldsymbol{\Omega}_I - \mathbf{y}'\| \quad (5.27)$$

3. Si d se encuentra dentro del umbral U establecido, el sistema decidirá que la muestra de es en de dicho individuo. Si d no se encuentra dentro del umbral establecido, el sistema decidirá que la muestra no pertenece al individuo en cuestión.

$$h(\mathbf{x}') = \begin{cases} 1, & \text{Si } d < U \\ 0, & \text{Si } d > U \end{cases} \quad (5.28)$$

Donde la decisión del sistema h en función del patrón de entrada \mathbf{x}' vale 1 si la imagen de entrada es clasificada positivamente para el individuo en cuestión y 0 en el caso contrario.

Finalmente, y si es de interés del lector, se comenta que Turk y Pentland también proponen un método de detección facial basado en *eigenfaces* [9]. A modo de resumen, su propuesta consiste en calcular la distancia que existe entre una imagen de entrada, y su proyección sobre el espacio de caras. Se considera importante destacar que combinan su técnica propuesta con detección de movimiento en un vídeo, sin embargo, podría combinarse con la detección de color de piel (por ejemplo) para imágenes fijas.

6. Análisis lineal discriminante (LDA)

El análisis por componentes principales es una herramienta muy útil para representar las muestras y reducir la dimensión de los datos, pero a la hora de clasificar existe la posibilidad de que no se comporte como es esperado. En la Figura 6.1 se puede observar uno de estos casos.

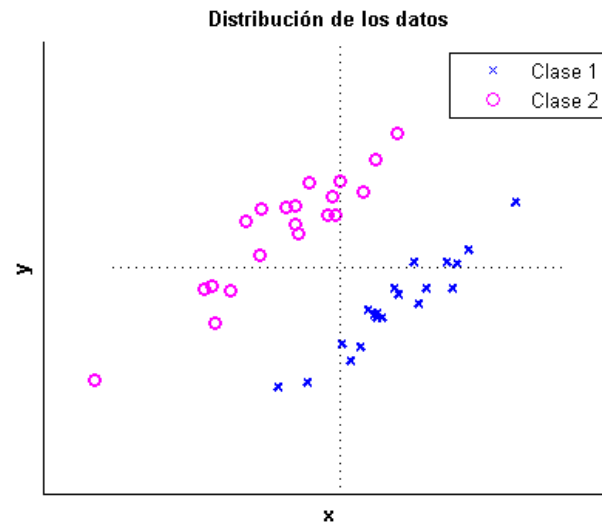


Figura 6.1 Diagrama de dispersión de dos clases cuyas muestras fueron generadas aleatoriamente en MATLAB.

Se trata de un conjunto de muestras generado aleatoriamente usando MATLAB. En él se observan 40 muestras de las cuales 20 corresponden a la clase 1 y las otras 20 corresponden a la clase 2. Aunque es prácticamente evidente cuál sería la dirección de la primera componente principal, en la Figura 6.2 se muestran ambas componentes principales (Figura 6.2.a). Se puede observar también que no existe una buena separabilidad entre clases una vez reducidas las dimensiones de los datos proyectándolos sobre la primera componente principal (Figura 6.2.b).

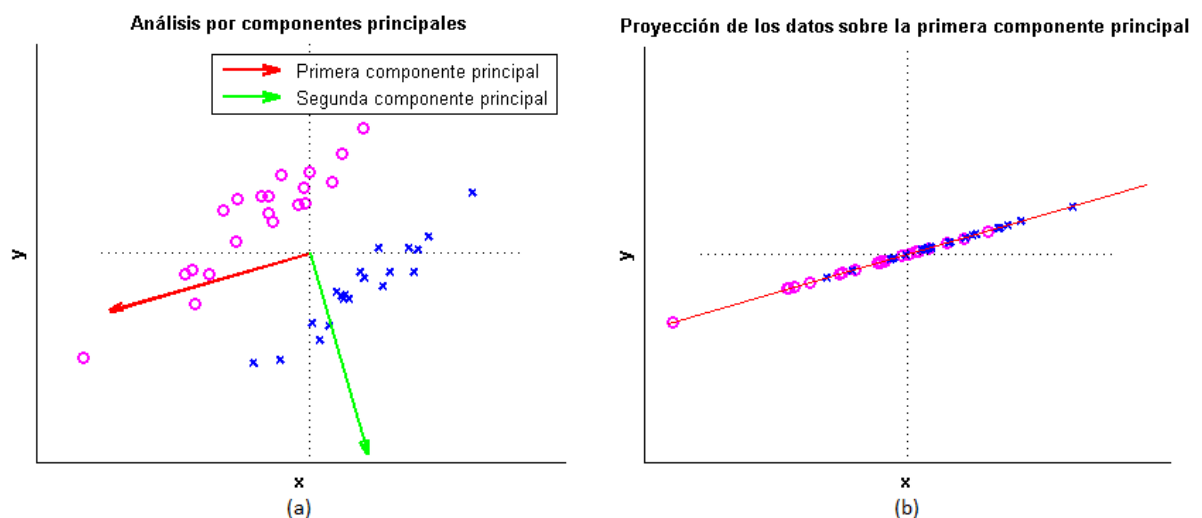


Figura 6.2. Análisis por componentes principales y posterior reducción de la dimensión de los datos de la Figura 6.1.

El objetivo del análisis lineal discriminante (LDA, por sus siglas en inglés), como su nombre indica, es encontrar una transformación que garantice una mejor discriminación entre clases. Esto se logra maximizando la distancia entre clases por un lado, y minimizando la varianza dentro de las clases por otro.

En realidad en este documento se utiliza el **Análisis lineal discriminante de Fisher** [22] (FLD, por sus siglas en inglés, *Fisher's Linear Discriminant*), pero los términos LDA y FLD se utilizarán indistintamente, siendo el FLD una de las posibles técnicas de Análisis Lineal Discriminante. Adicionalmente, para que los resultados del FLD sean óptimos se debe cumplir que la distribución de los datos sea normal y que las matrices de covarianzas sean iguales en todas las clases. No obstante, aunque no se cumplan del todo las condiciones, el Análisis Lineal Discriminante de Fisher arroja resultados que lo pueden calificar como un método robusto [23].

Maximizar la distancia entre clases es sencillo. Si se supone que de cada clase se tiene únicamente el vector media, entonces el eje que mejor representa los datos será el mismo que consiga maximizar la varianza de los datos al igual que en PCA. Esto es:

$$[S_B]_{n \times n} = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (6.1)$$

Donde S_B es la **matriz de covarianza entre clases**, C es el número de clases, N_i y μ_i el número de muestras y el vector media de cada clase i , respectivamente; y μ la media total de los datos. Dicho esto, los ejes que se buscan serían los autovectores de la matriz de covarianza entre clases. En la Figura 6.3 se puede observar la aplicación de este procedimiento sobre los datos de la Figura 6.1. Nótese como en este caso podría trazarse un umbral que separe ambas clases a partir de su proyección sobre la primera componente principal calculada exclusivamente con las medias de cada una.

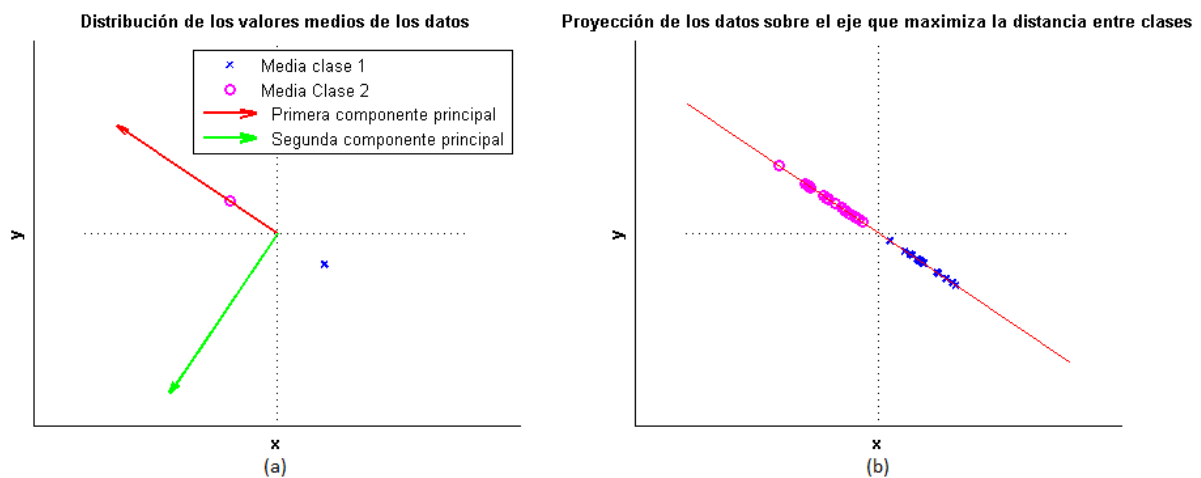


Figura 6.3. Demostración de la maximización de la distancia entre clases para las muestras de la Figura 6.1.

Para minimizar la varianza dentro de las clases, el procedimiento sería trasladar todas las distribuciones de las clases al origen de coordenadas y calcular los autovectores de la nueva matriz de covarianzas de esta nueva dispersión de los datos. Esto equivale a calcular para la siguiente matriz sus autovectores:

$$[S_W]_{n \times n} = \sum_{i=1}^C \sum_{\mathbf{x}_k \in X_i} (\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T \quad (6.2)$$

Donde se llama **matriz de covarianzas dentro de clases** a S_W , C es el número de clases, X_i el conjunto de muestras de la clase i , \mathbf{x}_k cada una de las muestras contenidas en dicho conjunto, y μ_i el vector media de la clase i . La aplicación de este procedimiento sobre los datos de la Figura 6.1 se observa en la Figura 6.4. Posteriormente, en lugar de elegir los autovectores correspondientes a los autovalores más altos, se eligen los correspondientes a los más bajos. Obsérvese que para este caso se consigue una separación entre clases mucho mayor que para el caso anterior.

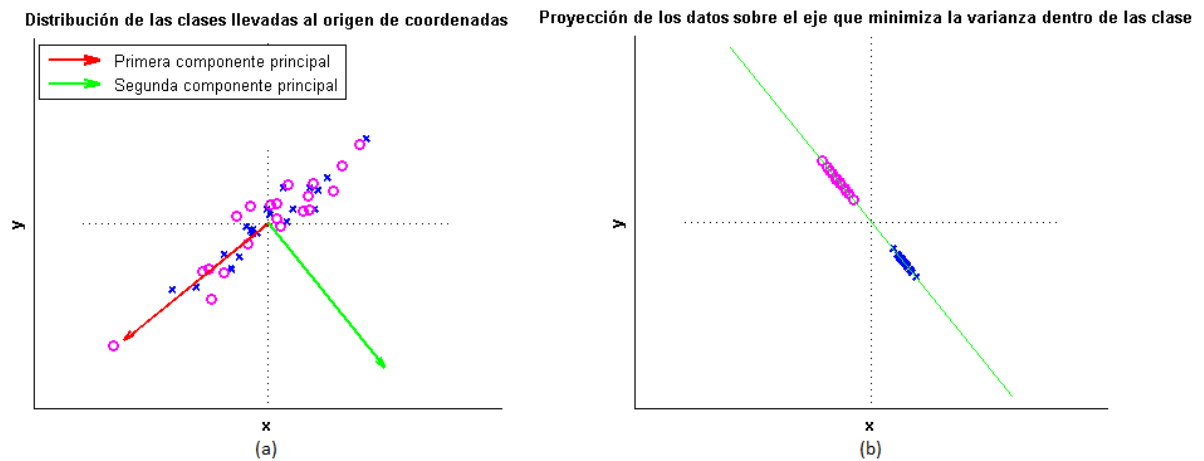
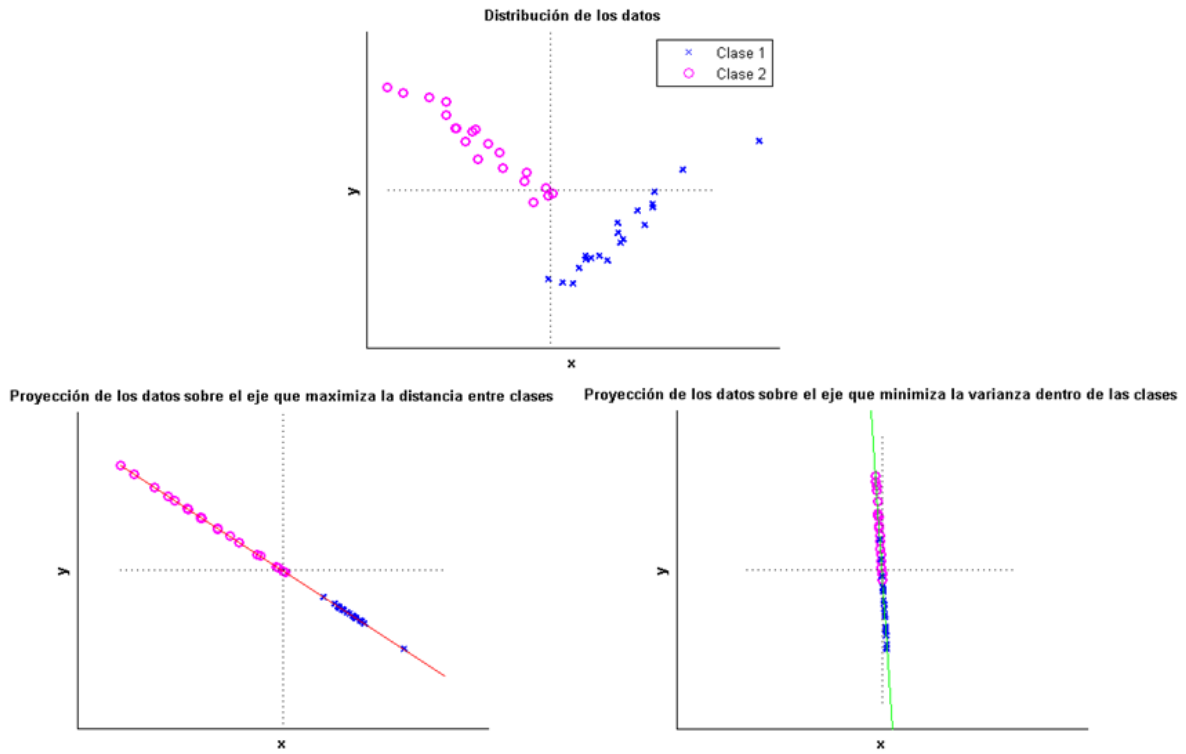


Figura 6.4. Minimización de la varianza dentro de las clases para las muestras de la Figura 6.1.

Si bien para la distribución de los datos de la Figura 6.1 parece que no cambiase mucho el resultado entre solamente maximizar la distancia entre clases (Figura 6.3) o minimizar la varianza dentro de las clases (Figura 6.4), no siempre es así. Para demostrarlo, en la Figura 6.5 se observan dos casos de distribuciones donde no es indiferente una que la otra. En el caso A las clases quedan discriminadas al reducir la dimensión según el eje que maximiza la distancia entre datos, pero no es así al reducir la dimensión según el eje que minimiza la varianza dentro de los datos. El caso B es el contrario al A.

CASO A



CASO B

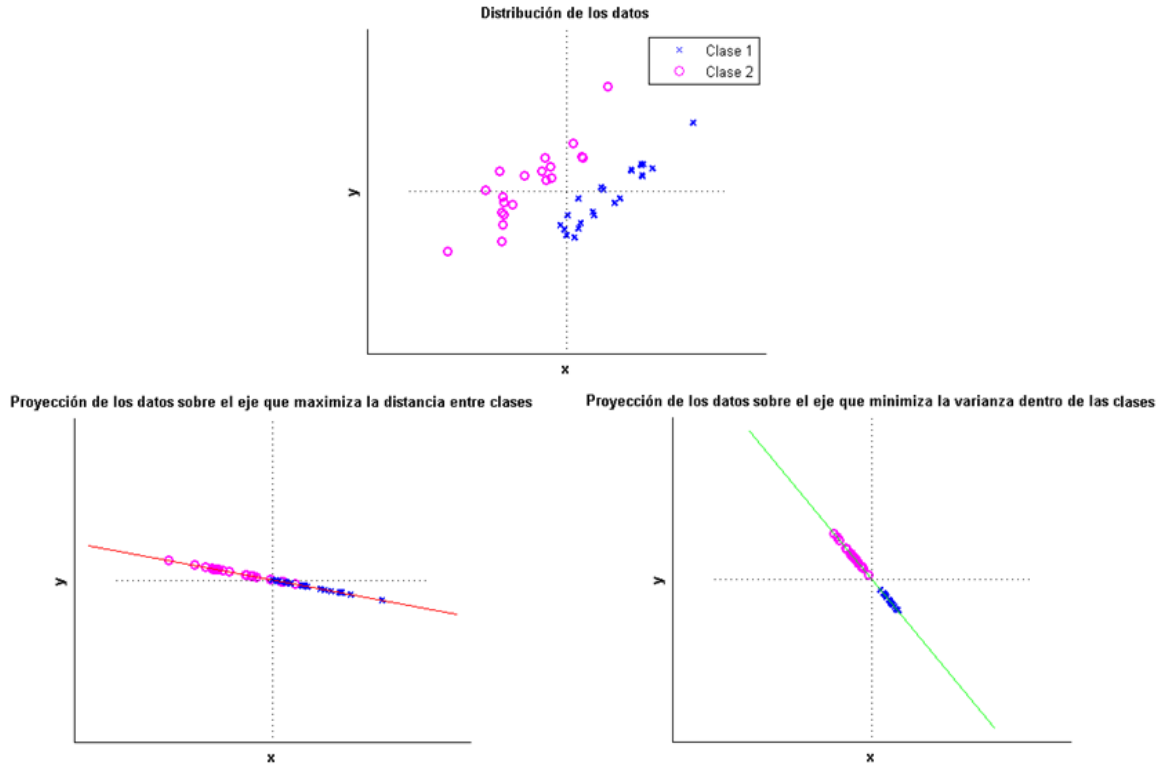


Figura 6.5. Casos de reducción de dimensión según el eje que maximiza la distancia entre clases y según el eje que minimiza la varianza dentro de clases. Al igual que las figuras anteriores, tanto la distribución aleatoria de las muestras, como sus proyecciones sobre los ejes indicados fueron generadas en MATLAB.

Para poder calcular los vectores que forman el subespacio que maximiza la distancia entre clases y a la vez minimiza la varianza dentro de las clases hay que tomar en cuenta que al calcular los autovectores de S_W^{-1} , los correspondientes a los autovalores más altos son los mismos que los correspondientes a los autovalores más bajos de S_W . Por lo tanto, para conseguir el objetivo propuesto hay que calcular los vectores propios y valores propios de la matriz $S_W^{-1}S_B$. Esto sería equivalente a calcular los autovectores y autovalores que satisfagan la ecuación:

$$S_W^{-1}S_B\mathbf{w}_i = \lambda_i\mathbf{w}_i \rightarrow S_B\mathbf{w}_i = \lambda_i S_W\mathbf{w}_i \quad (6.3)$$

Dichos autovectores resultantes serán las columnas que formarán la matriz P , que de igual forma que en el análisis por componentes principales, será utilizada para proyectar los datos en el espacio propio calculado. Es importante tomar en cuenta que el número de autovectores resultantes distintos de cero será igual al número de clases menos uno ($C - 1$) [24].

En la Figura 6.6 se observa un ejemplo de análisis lineal discriminante para 3 clases. Al igual que todos los ejemplos mostrados hasta ahora, fue generado aleatoriamente usando MATLAB de manera que cada clase siga una distribución normal. Nótese que el LDA garantiza una mejor separabilidad entre clases que el PCA.

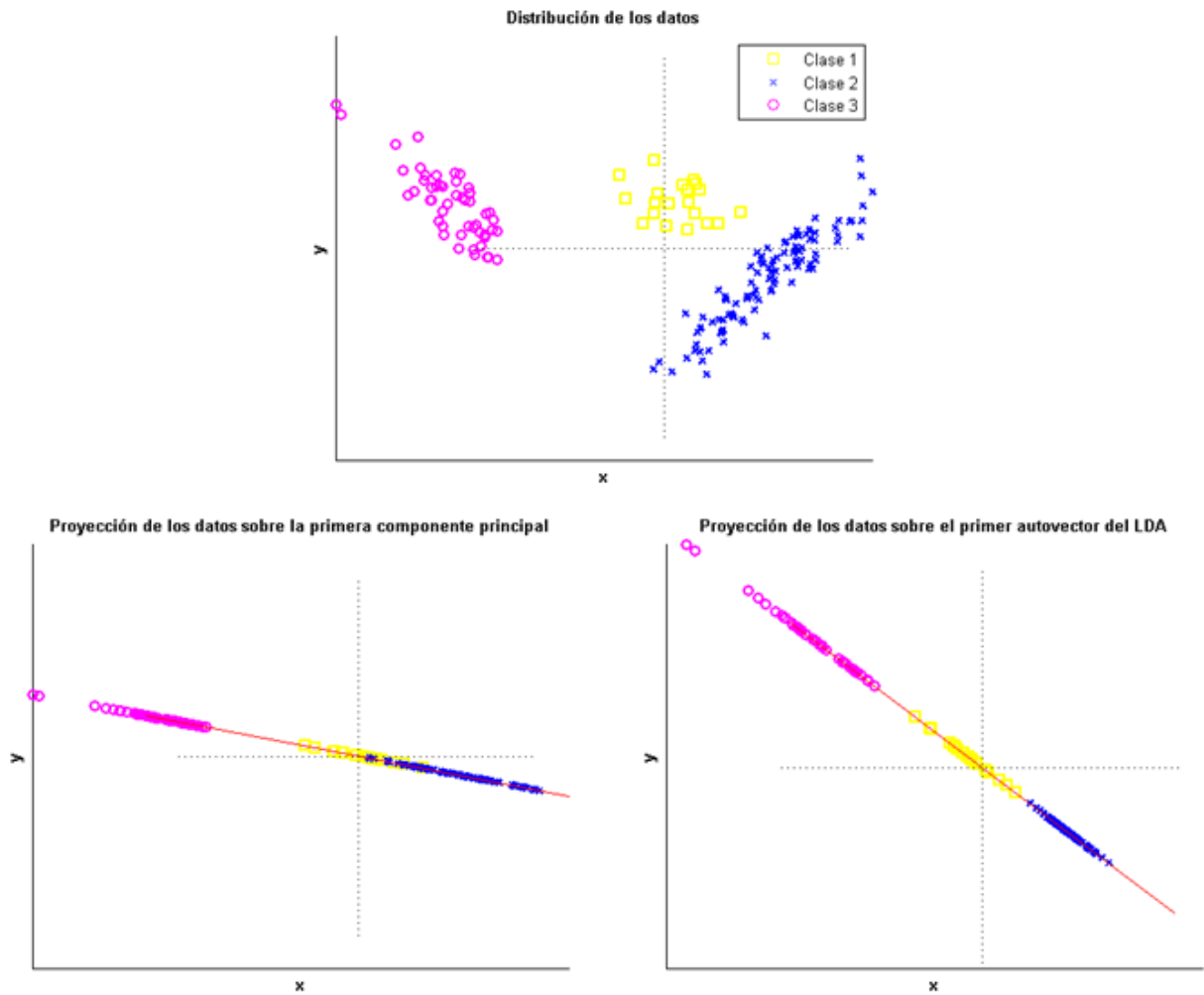


Figura 6.6. Ejemplo de una distribución de 3 clases donde se comprueba que el LDA garantiza mayor separabilidad entre clases que el PCA.

6.1. Reconocimiento facial usando *fisherfaces*

Una vez estudiado el análisis lineal discriminante y comparándolo con el análisis por componentes principales, aplicarlo para el reconocimiento facial es una tarea sencilla.

Sin embargo, para un set de entrenamiento X formado por N imágenes de C individuos (clases), surge el problema de que la matriz de covarianza dentro de clases es siempre singular (no invertible). Esto se debe al hecho de que su rango es como mucho $N - C$, y generalmente el número de imágenes de entrenamiento es mucho menor que el número de píxeles n de una imagen (Recordar que $S_W \in R^n$).

El procedimiento que proponen Belhumeur et al. [22] es usar el análisis por componentes principales para reducir la dimensión de los datos hasta $N - C$, y posteriormente aplicar FLD para reducir la dimensión hasta $C - 1$, que es el número de autovectores distintos de cero resultantes de (6.3) [24].

Dicho esto, los vectores columna que formarían a la matriz P en (5.12) serían:

$$P = P_{PCA}P_{LDA} \rightarrow P^T = P_{LDA}^T P_{PCA}^T \quad (6.4)$$

De igual manera que a los autovectores resultantes del PCA se les llama *eigenfaces*, a este nuevo conjunto de vectores propios se les suele llamar *fisherfaces*.

A continuación, el reconocimiento (precedido por una etapa de entrenamiento donde se recogen los patrones de los individuos a clasificar) se realiza de igual forma que con las *eigenfaces*, proyectando la imagen de entrada en el subespacio formado por las *fisherfaces* y comparándolas con los patrones registrados. La siguiente fórmula indica cómo calcular las coordenadas \mathbf{y} de una imagen de entrada \mathbf{x} en el subespacio formado por las *fisherfaces*:

$$\mathbf{y} = P^T \mathbf{x} = P_{LDA}^T P_{PCA}^T \mathbf{x} \quad (6.5)$$

En la Figura 6.7 se observan las *fisherfaces* calculadas en MATLAB correspondientes al set de entrenamiento de la Figura 5.4. Si son comparadas con las *eigenfaces* (Figura 5.6) es interesante ver cómo éstas tienen un color gris más uniforme que las primeras *eigenfaces*. Esto podría atribuirse a que las *eigenfaces* fueron generadas para maximizar la varianza total de las imágenes de entrenamiento, mientras que las *fisherfaces* fueron generadas con la intención de crear mayor discriminación entre clases.



Figura 6.7. Fisherfaces del set de entrenamiento de la Figura 5.4.

7. Wavelets de Gabor

7.1. El análisis multirresolución y las *wavelets* en general

Cuando se calcula la transformada de Fourier de una señal para hallar su espectro, se obtienen las componentes sinusoidales de esa señal (información en frecuencia). Sin embargo, no se obtiene información sobre las tramas de la señal en que se encuentran dichas componentes. Por ello, en caso de necesitar esta información se utiliza la transformada localizada de Fourier o transformada de Fourier de tiempo corto (STFT, por sus siglas en inglés), que consiste básicamente en multiplicar la señal por una función ventana ubicada al principio de la señal y calcular la transformada de Fourier a esa porción y así a lo largo de toda la señal.

Como es evidente que explicar cómo se calcula la transformada de Fourier no es el objetivo de este documento, para más información se recomienda la siguiente fuente: [25].

Para continuar, se recuerdan las ecuaciones de la transformada de Fourier y la transformada localizada de Fourier:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (7.1)$$

$$STFT\{f(t)\}(\tau, \omega) = F(\tau, \omega) = \int_{-\infty}^{\infty} f(t)w(t - \tau)e^{-j\omega t} dt \quad (7.2)$$

Donde $F(\omega)$ y $F(\tau, \omega)$ son los coeficientes de Fourier para cada tipo de transformada, y $w(t - \tau)$ es la función ventana de la transformada localizada de Fourier. Se considera importante destacar que aunque las fórmulas que en este capítulo se presentan son para tiempo continuo, a lo largo de este proyecto realidad se trabaja con tiempo discreto.

Como se puede observar, de la transformada de Fourier solamente se obtiene un coeficiente por frecuencia. Esto hace que la información de la señal en el tiempo se pierda completamente, pero que en frecuencia la señal quede completamente definida. Sin embargo, no se sabe en qué tramas de la señal se encuentran dichas componentes de frecuencia (a menos que se trate de una señal estacionaria, en cuyo caso las componentes de frecuencia están presentes todo el tiempo en la señal).

Por otro lado, aunque por definición la integral se extienda a todo el tiempo, los coeficientes de la STFT se restringen a la duración de la función ventana, que es desplazada por el parámetro τ a lo largo del tiempo para obtener la información de frecuencia localizada en el tiempo.

No obstante, la desventaja que presenta STFT es un problema de resolución. Para ventanas estrechas (análisis de banda ancha), las componentes de frecuencia baja no son

detectadas en el espectro, dando como resultado baja resolución en frecuencia para dichas frecuencias (se obtiene información sobre ciertas bandas de frecuencia, más no se obtiene información detallada sobre dichas frecuencias). Por otro lado, si se ensancha el tamaño de la ventana (análisis de banda estrecha), se obtiene mejor resolución en frecuencia, pero se pierde localización en el tiempo (si se ensancha el tamaño de la ventana hasta abarcar la longitud temporal de la señal el resultado será el mismo que con la transformada de Fourier). En consecuencia, como la duración de la ventana es fija, la resolución del análisis de la señal tanto en tiempo, como en frecuencia es fija para la transformada localizada de Fourier. En la Figura 7.1 se describe de manera gráfica la comparación entre la resolución en tiempo y la resolución en frecuencia para la transformada de Fourier y la STFT.

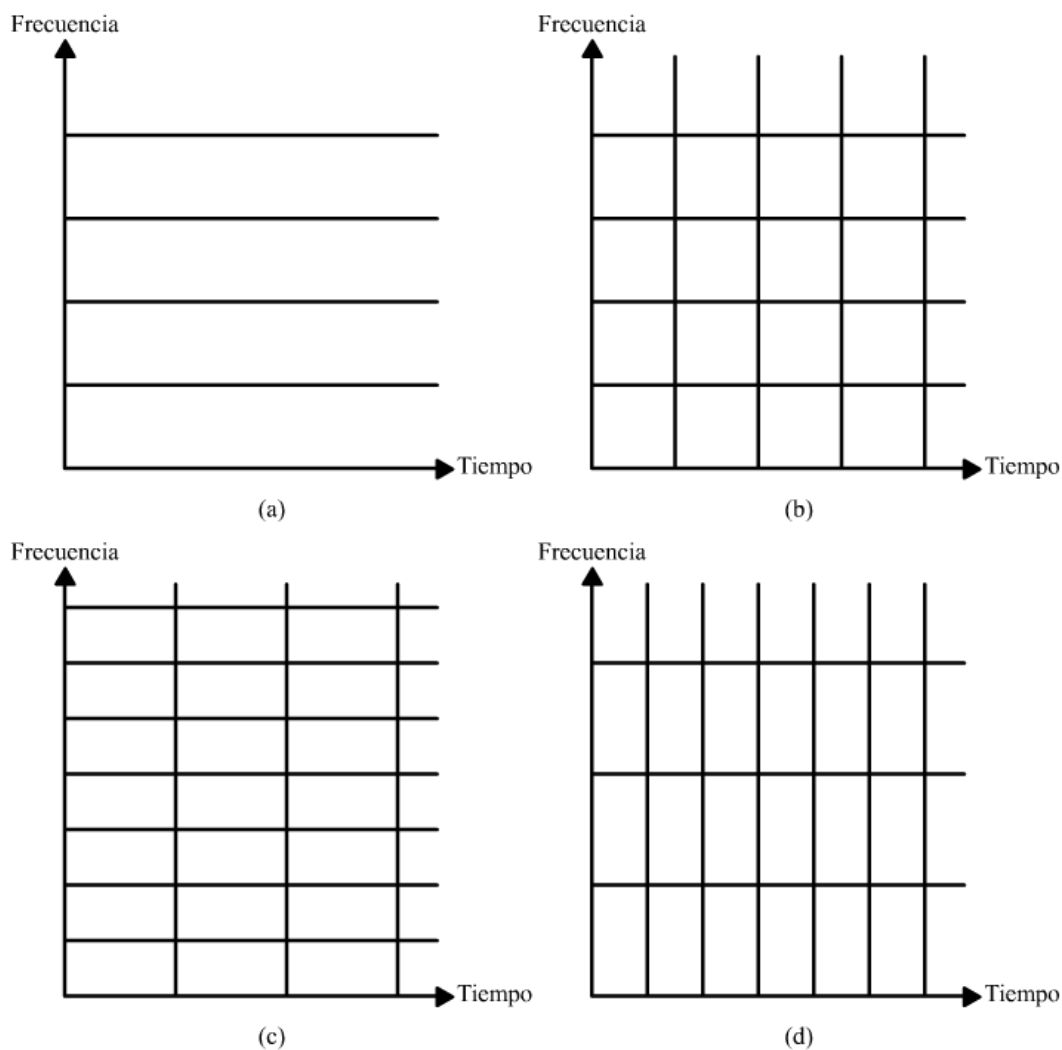


Figura 7.1. Relación entre resolución frecuencial y temporal para (a) la transformada de Fourier, (b) para una STFT con tamaño de ventana medio, (c) para una STFT de banda estrecha (ventana ancha) y (d) para una STFT de banda ancha (ventana estrecha).

A raíz de esto se empezó a indagar sobre un enfoque diferente que para componentes de baja frecuencia obtenga alta resolución en frecuencia (baja resolución en tiempo), y que para componentes de alta frecuencia obtenga alta resolución en tiempo (baja resolución en frecuencia). Esto se debe a que por naturaleza, en aplicaciones prácticas, las bajas frecuencias suelen estar presentes durante más tiempo en una señal, y las altas

frecuencias suelen estar presentes durante menos tiempo. De esta forma se originó el **análisis multirresolución**. La idea de la relación resolución en tiempo-resolución en frecuencia del análisis multirresolución se observa en la Figura 7.2.

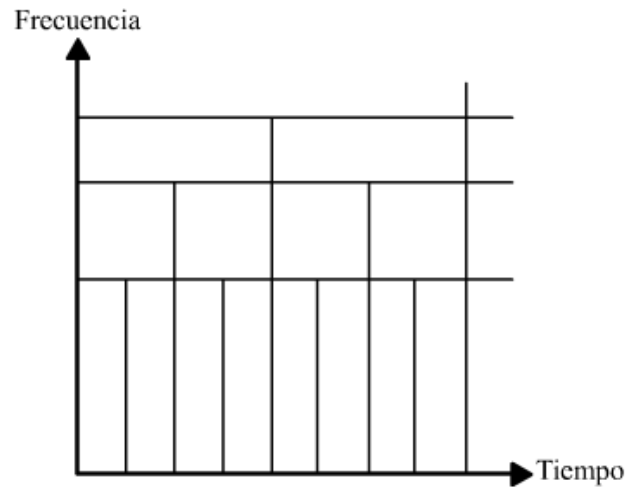


Figura 7.2. Relación entre resolución frecuencial y temporal para el análisis multirresolución.

Desde el punto de vista del álgebra lineal, la ecuación de la transformada de Fourier equivaldría a calcular la proyección de una señal sobre una base, que este caso sería la exponencial compleja a distintas frecuencias. En consecuencia, los coeficientes resultantes equivaldrían a las coordenadas de la señal sobre dicha base.

En el caso del análisis multirresolución, se utilizan como base de la proyección las conocidas como **wavelets**, también llamadas **ondículas** u **ondeletas**. Por este motivo, a la proyección de una función sobre una familia de **wavelets** se le suele llamar **transformada wavelet**. El término proviene del francés *ondelette*, que significa “pequeña onda” y son ondas cuya amplitud comienza en cero, crece hasta cierto valor y luego descienden nuevamente a cero.

Se utiliza el término **wavelet madre** $\psi(t)$ para referirse a aquella **wavelet** que es escalada y desplazada en el tiempo para formar una familia de **wavelets**. Por definición, las **wavelets** madre deben tener energía unidad (para mantener la energía de una señal al proyectarla) y valor medio nulo (No debe tener componente continua) [26]. Para formar una **wavelet** a partir de una **wavelet** madre se utiliza la ecuación:

$$\psi_{s,\tau} = \frac{1}{\sqrt{s}} \psi\left(\frac{t - \tau}{s}\right) \quad (7.3)$$

Donde ψ es la **wavelet** madre, s es la escala aplicada y τ el desplazamiento.

En la Figura 7.3 se observa un ejemplo particular de **wavelet** llamado “**Wavelet de Ricker**” y se trata de la segunda derivada negativa y normalizada de una función gaussiana. En la Figura 7.4 se muestra además la misma **wavelet**, pero a diferentes escalas. Nótese que al aumentar la escala, la anchura de la **wavelet** disminuye, y viceversa, por lo tanto, el parámetro s es inversamente proporcional a la frecuencia.

$$\psi(t) = \frac{2}{\sqrt{3\sigma\pi^{\frac{1}{4}}}} \left(1 - \frac{t^2}{\sigma^2}\right) e^{\frac{-t^2}{2\sigma^2}}$$

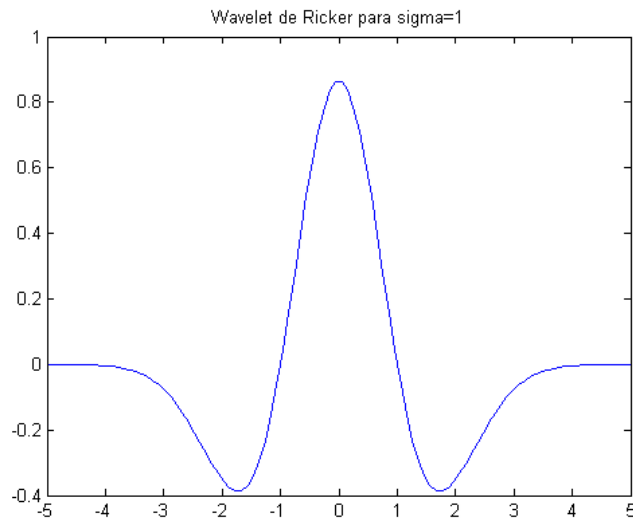


Figura 7.3. Wavelet de Ricker. También conocida como Mexican hat wavelet (wavelet sombrero mexicano).

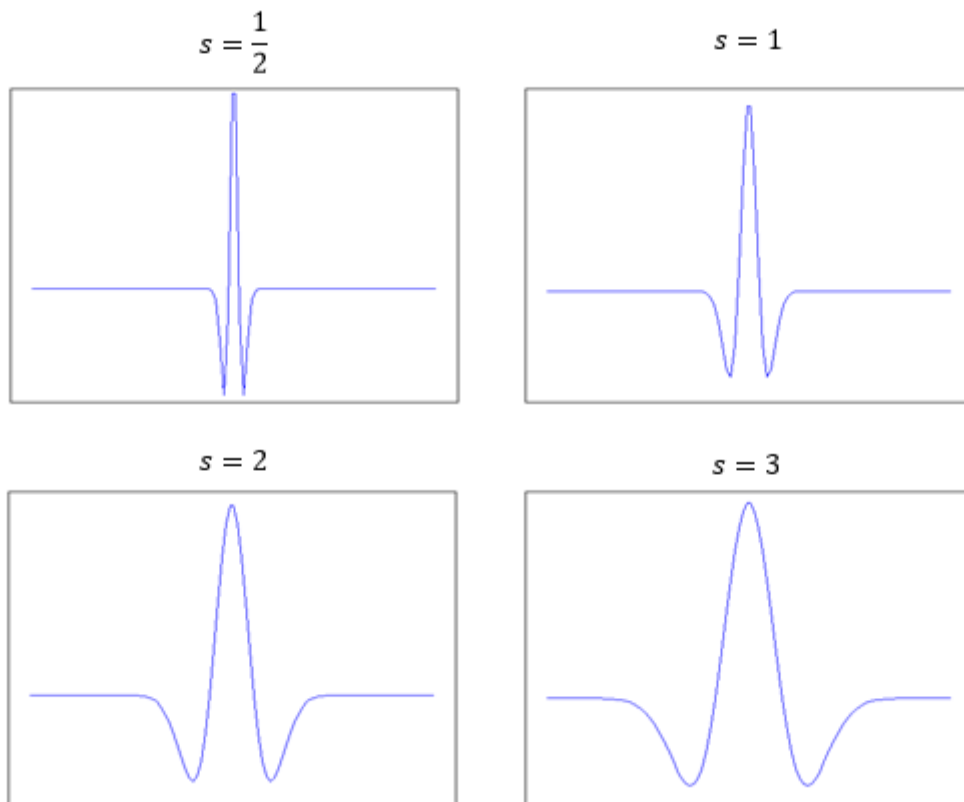


Figura 7.4. Wavelet de Ricker para diferentes escalas. La escala $s = 1$ es la wavelet madre. Todas las gráficas fueron generadas en MATLAB.

Es importante comentar que la teoría sobre las *wavelets* es mucho más extensa de lo que aquí se expone, pero esta pequeña introducción es suficiente para la aplicación que se les dará más adelante en este proyecto. Para más información, se recomiendan las siguientes fuentes: [26] [27] [28].

7.2. Filtros de Gabor

Los filtros de Gabor son filtros lineales cuya respuesta al impulso es una portadora sinusoidal multiplicada por una envolvente gaussiana.

Para una señal unidimensional, un filtro de Gabor puede servir como un filtro paso-banda puesto que su espectro es una gaussiana centrada a la frecuencia de la senoide que lo forma [29]. No obstante, en el entorno de este proyecto se trabaja con imágenes, por lo que se utilizarán filtros de Gabor bidimensionales (que luego se comprobará que también tienen el comportamiento de un filtro paso banda).

La fórmula de un filtro de Gabor bidimensional corresponde con la siguiente:

$$g(x, y) = s(x, y) w(x, y) \quad (7.4)$$

Donde s es la señal sinusoidal compleja y w es a la envolvente gaussiana.

De forma general, la función sinusoidal se define como:

$$s(x, y) = e^{j(2\pi(u_0x+v_0y)+P)} \quad (7.5)$$

Donde u_0 y v_0 son las frecuencias espaciales (en píxeles por ciclo) y P la fase de la señal. Al ser la senoide una función exponencial compleja, ésta puede ser dividida en su parte real e imaginaria

Los parámetros u_0 y v_0 también pueden ser representados en coordenadas polares:

$$\begin{aligned} F_0 &= \sqrt{u_0^2 + v_0^2} \\ \omega_0 &= \tan^{-1} \frac{v_0}{u_0} \end{aligned} \quad (7.6)$$

Y de forma inversa:

$$\begin{aligned} F_0 &= \sqrt{u_0^2 + v_0^2} \\ \omega_0 &= \tan^{-1} \frac{v_0}{u_0} \end{aligned} \quad (7.7)$$

Con lo que la senoide quedaría:

$$s(x, y) = e^{j(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P)} \quad (7.8)$$

En la Figura 7.5 se puede observar la parte real e imaginaria de dos funciones sinusoidales con diferentes parámetros. Es importante tomar en cuenta que se trata de imágenes de 128x128 píxeles, puesto que este es el motivo por el que para la primera exponencial compleja ($u_0 = \frac{1}{64}, v_0 = 0$) sólo se observan 2 ciclos, ya que las unidades de la frecuencia son $\frac{\text{ciclos}}{\text{pixel}}$. Nótese además que al tener polaridad nula y haber escogido una frecuencia convenientemente baja, se observa claramente la forma de coseno de la parte real y la forma de seno de la parte imaginaria.

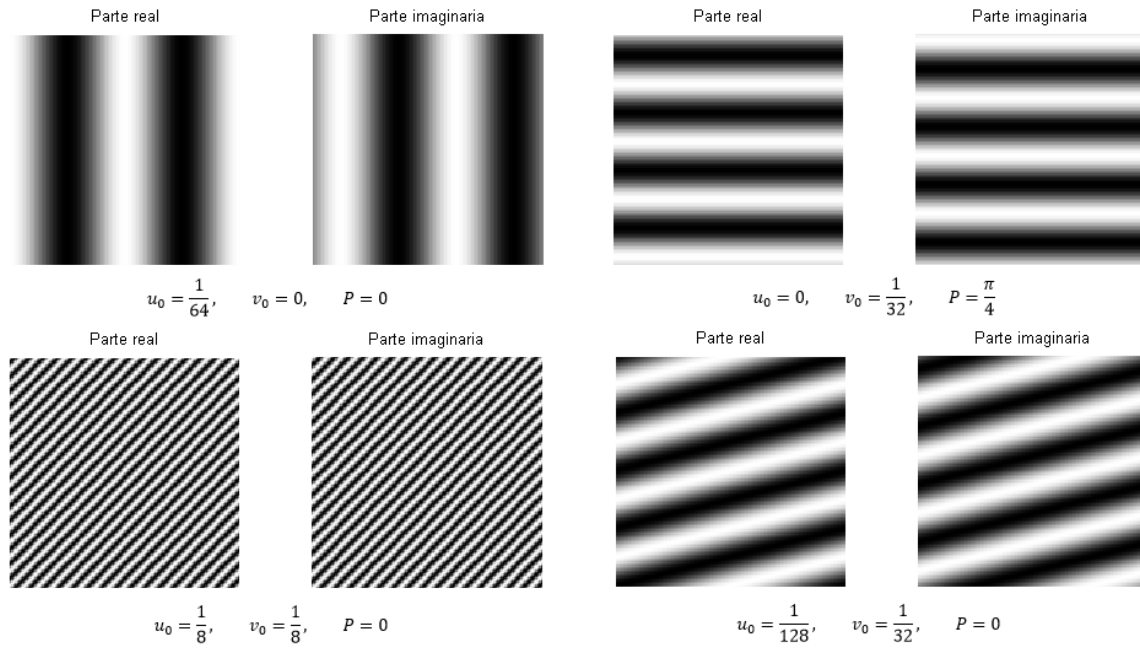


Figura 7.5. Gráficas de 4 exponenciales complejas bidimensionales con diferentes parámetros.

La forma de la envolvente gaussiana de un filtro de Gabor corresponde con la siguiente fórmula:

$$w(x, y) = K e^{-\frac{1}{2}(\frac{x_p^2}{\sigma_x^2} + \frac{y_p^2}{\sigma_y^2})} \quad (7.9)$$

$$x_p = (x - x_0) \cos \theta + (y - y_0) \sin \theta$$

$$y_p = -(x - x_0) \sin \theta + (y - y_0) \cos \theta$$

Donde K es una constante para escalar la magnitud de la gaussiana, σ_x y σ_y son las desviaciones estándar que controlan la anchura de la campana gaussiana en las direcciones de los ejes x y y , correspondientemente, (x_0, y_0) son las coordenadas del centro de la gaussiana y θ el ángulo de rotación de la gaussiana.

En la Figura 7.6 se pueden observar la parte real e imaginaria de dos funciones gaussianas con diferentes parámetros.

$$x_0 = -32, \quad y_0 = -32, \quad \sigma_x = 16, \quad \sigma_y = 16 \qquad \sigma_x = 16, \quad \sigma_y = 32, \quad \theta = \frac{\pi}{4}$$

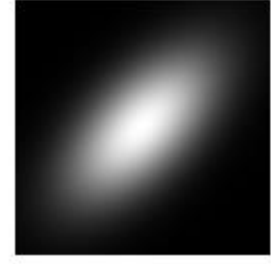


Figura 7.6. Gráfica de dos envolventes gaussianas con distintos parámetros. Los parámetros indicados en la primera gaussiana son tomando en cuenta que el origen de coordenadas está en el centro de la imagen.

Aunque no se indique, la segunda gaussiana está centrada en las coordenadas (0,0).

Finalmente, la fórmula de un filtro de Gabor correspondería con la siguiente:

$$g(x, y) = K e^{-\frac{1}{2}(\frac{x_p^2}{\sigma_x^2} + \frac{y_p^2}{\sigma_y^2})} e^{j(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P)} \quad (7.10)$$

En lo que a este proyecto concierna, la fórmula de los filtros de Gabor se puede simplificar tomando en cuenta las siguientes consideraciones:

- 1- La fase de la senoide es nula ($P = 0$).

$$s(x, y) = e^{j(2\pi F_0(x \cos \omega_0 + y \sin \omega_0))} \quad (7.11)$$

- 2- La relación de aspecto $\frac{\sigma_x}{\sigma_y}$ de la gaussiana es igual a la unidad ($\sigma_x = \sigma_y = \sigma$).

$$w(x, y) = K e^{-\frac{1}{2}(\frac{x_p^2 + y_p^2}{\sigma^2})} \quad (7.12)$$

- 3- El origen de la senoide es el mismo que el de la envolvente gaussiana.

$$s(x, y) = e^{j(2\pi F_0((x-x_0) \cos \omega_0 + (y-y_0) \sin \omega_0))} \quad (7.13)$$

- 4- La rotación de la frecuencia espacial F_0 resultado de la superposición de la frecuencias espaciales u_0 y v_0 es igual a la rotación de la gaussiana ($\omega_0 = \theta$).

$$\begin{aligned} s(x, y) &= e^{j(2\pi F_0((x-x_0) \cos \theta + (y-y_0) \sin \theta))} \\ &= e^{j(2\pi F_0 x_p)} \end{aligned} \quad (7.14)$$

Finalmente, la fórmula del filtro de Gabor simplificada quedaría:

$$g(x, y) = s(x, y) w(x, y) = K e^{-\frac{1}{2} \left(\frac{x_p^2 + y_p^2}{\sigma^2} \right)} e^{j(2\pi F_0 x_p)} \quad (7.15)$$

En Figura 7.7 se muestran 3 filtros de Gabor calculados en MATLAB a partir de la fórmula simplificada de este proyecto, cuyos parámetros son distintos para cada uno. Las imágenes generadas son de 256x256 píxeles. Nótese que para Figura 7.7.b y Figura 7.7.c, el espacio cubierto por la envolvente gaussiana es el mismo, mientras que la frecuencia es más alta en Figura 7.7.c. Por otro lado, la relación existente entre σ y F_0 en la Figura 7.7.a y Figura 7.7.b es la misma (σ es la mitad de la longitud de onda), de manera que el número de ciclos que cubre la envolvente gaussiana es el mismo para ambos casos. Esto último tendrá especial importancia en el siguiente apartado sobre *wavelets* de Gabor.

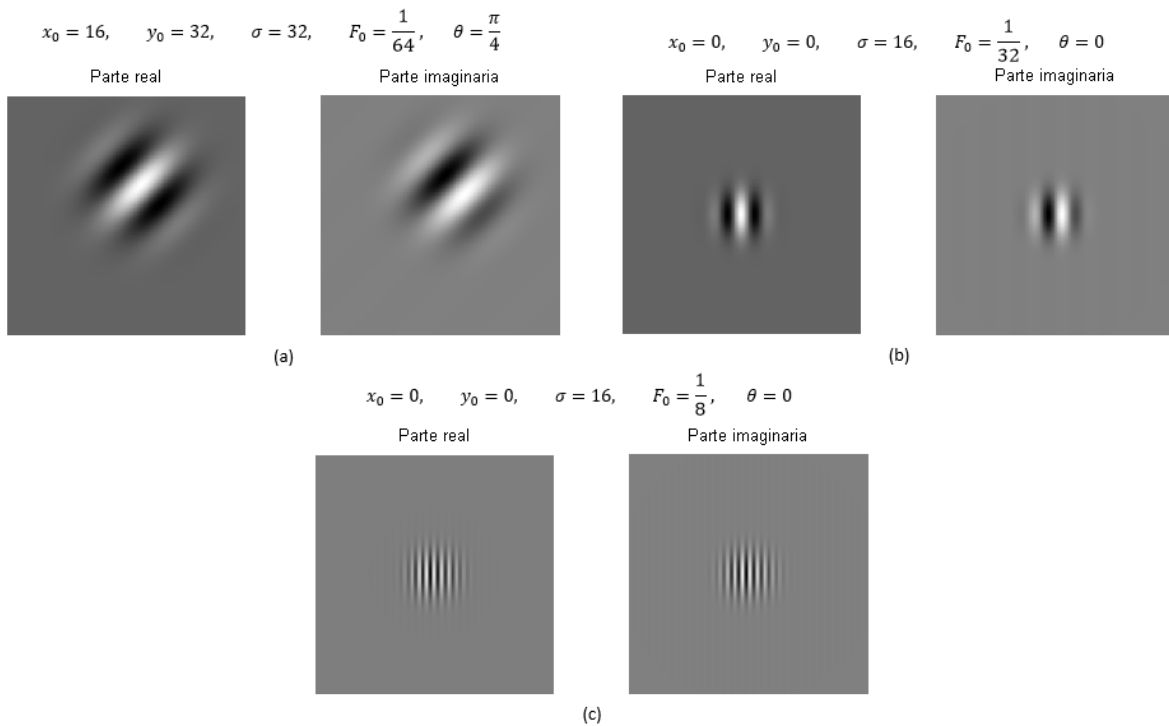


Figura 7.7. Filtros de Gabor con distintos parámetros. El centro de coordenadas está en el origen de la imagen.

7.3. Las *wavelets* de Gabor

Como se puede suponer de la Figura 7.7, los filtros de Gabor no son *wavelets* por tres motivos:

- 1- El primer motivo es evidente, y es que cada filtro (de la misma rotación, por supuesto) no es una escala de los otros. Se recuerda que esto es una propiedad fundamental de las *wavelets*, ya que de forma contraria los filtros se comportarían con una resolución fija tanto en el espacio, como en la frecuencia.
- 2- Aunque los otros dos motivos no sean muy evidentes, uno de ellos puede observarse literalmente en la Figura 7.7. Se trata de la componente continua. Los filtros de Gabor en realidad son funciones con valores muy pequeños para poder ser

visualizados en una imagen, por lo tanto se les aplica una expansión para que sus valores se extiendan, por ejemplo, desde cero hasta 255 (Valores mínimo y máximo de un píxel para una imagen de 8 bits). Si el filtro tenía inicialmente una componente continua, ésta será expandida con el resto de la señal y podrá observarse, siendo este el motivo por el que el gris de fondo de los filtros de la Figura 7.7.a y la Figura 7.7.b se ven más oscuros con respecto al filtro Figura 7.7.c.

- 3- El último motivo no puede deducirse de la Figura 7.7, y es que la energía de un filtro de Gabor no es la unidad, a menos que sea corregido cambiando el valor de la constante K que multiplica a la función gaussiana.

No obstante, los filtros de Gabor se pueden comportar como *wavelets* corrigiendo los tres inconvenientes comentados.

Como se podía deducir de la Figura 7.7, para cumplir que un filtro sea escala de otro (de misma θ) es necesario que la envolvente gaussiana cubra siempre la misma longitud de onda, por lo tanto se modifica la fórmula para que la desviación típica de la gaussiana tenga dependencia de frecuencia [30]:

$$\sigma = \frac{1}{F_0} \quad (7.16)$$

En la Figura 7.8 se puede observar el efecto de la dependencia con la frecuencia en σ para una *wavelet* de Gabor unidimensional. Al lado izquierdo de la figura se ven los filtros cuyas desviaciones típicas no dependen de la frecuencia y por tanto la envolvente gaussiana es la misma para todas las frecuencias de la senoide. Al lado derecho se observa que al depender la desviación típica de la frecuencia, la envolvente gaussiana cubre siempre el mismo número de longitudes de onda, haciendo que cada frecuencia sea escala de las otras. Para las *wavelets* de Gabor la envolvente gaussiana puede verse como una ventana de longitud variable en función de la frecuencia, ya que si fuera de longitud fija, se estaría aplicando una STFT en lugar de una transformada *wavelet*.

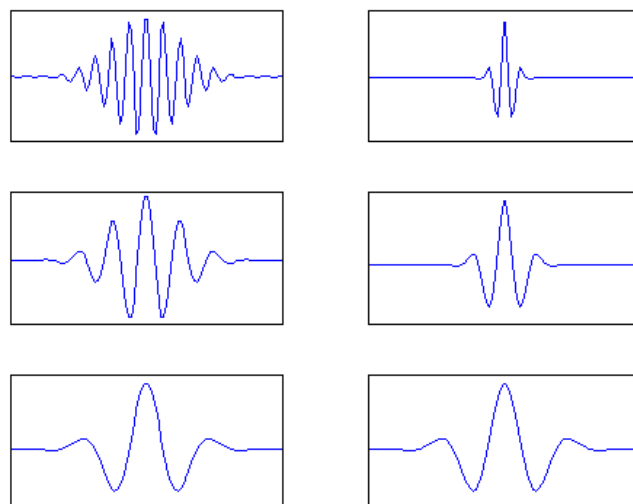


Figura 7.8. A la izquierda se observan filtros de Gabor unidimensionales a distintas frecuencias con σ fijo, y a la derecha otros filtros con las mismas frecuencias que los de la izquierda, pero con σ dependiente de F_0 .

Otros autores [31] utilizan el vector de onda $\vec{k} = 2\pi F_0 e^{j\omega_0} = 2\pi F_0 e^{j\theta}$ y cambian el término σ^2 por $\frac{\sigma^2}{k^2}$ para definir la fórmula de una *wavelet* de Gabor de manera más general, pero al final la fórmula acaba siendo la misma (porque además suelen utilizar $\sigma = 2\pi$). A efectos de este proyecto; como es sabido, \vec{k} es paralelo a la dirección de la onda, y ésta corresponde con la dirección de la componente x_p (recordar que $\omega_0 = \theta$), por lo tanto se va a utilizar solamente su magnitud, el número de onda $k = 2\pi F_0$, para definir la fórmula de las *wavelet* de Gabor, simplificándola.

$$\psi(\vec{x}) = K e^{-\frac{1}{2}(\frac{k^2 x^2}{\sigma^2})} e^{j(\vec{k}\vec{x})} \rightarrow \psi(x, y) = K e^{-\frac{k^2}{2}(\frac{x_p^2 + y_p^2}{\sigma^2})} e^{j(kx_p)} \quad (7.17)$$

En segundo lugar se va a tratar la energía del filtro para que sea igual a la unidad. Para conseguirlo, se normaliza el filtro con respecto a su energía haciendo que la constante K de la gaussiana del filtro sea igual a uno partido de su energía [32].

$$\psi(x, y) = \frac{k}{\sqrt{\pi}\sigma} e^{-\frac{k^2}{2}(\frac{x_p^2 + y_p^2}{\sigma^2})} e^{j(kx_p)} \quad (7.18)$$

Para terminar, una *wavelet* no debe tener componente continua, y aunque el seno (parte imaginaria) de la función exponencial compleja no introduce componente continua al ser una función impar, el coseno (parte real) sí, porque es una función par, y al multiplicarla por los bordes de la envolvente gaussiana, introduce una componente continua. En consecuencia, hay que restarle a la función sinusoidal la transformada de Fourier de la *wavelet* para $u_0 = 0$ y $v_0 = 0$ [31] [32], resultando la fórmula final de las *wavelets* de Gabor:

$$\psi(x, y) = \frac{k}{\sqrt{\pi}\sigma} e^{-\frac{k^2}{2}(\frac{x_p^2 + y_p^2}{\sigma^2})} \left[e^{j(kx_p)} - e^{-\frac{\sigma^2}{2}} \right] \quad (7.19)$$

Cada familia de *wavelets* de Gabor bidimensionales se forma combinando una rotación θ con distintas frecuencias F_0 (escalas). En la Figura 7.9 se observa la parte real de 8 familias de *wavelets* de Gabor cada una con frecuencias: $\frac{1}{4} \frac{\text{ciclos}}{\text{pixel}}$, $\frac{1}{4\sqrt{2}} \frac{\text{ciclos}}{\text{pixel}}$, $\frac{1}{8} \frac{\text{ciclos}}{\text{pixel}}$, $\frac{1}{8\sqrt{2}} \frac{\text{ciclos}}{\text{pixel}}$ y $\frac{1}{16} \frac{\text{ciclos}}{\text{pixel}}$. En Figura 7.10 se observa la forma tridimensional de una *wavelet* de Gabor con los valores reales de cada uno de sus píxeles, en lugar de las versiones expandidas que se observan en las imágenes. Obsérvese dónde está ubicado el origen de coordenadas de dicha gráfica. Esto significa que las *wavelets* en realidad están centradas en $(x_0, y_0) = (64, 64)$.

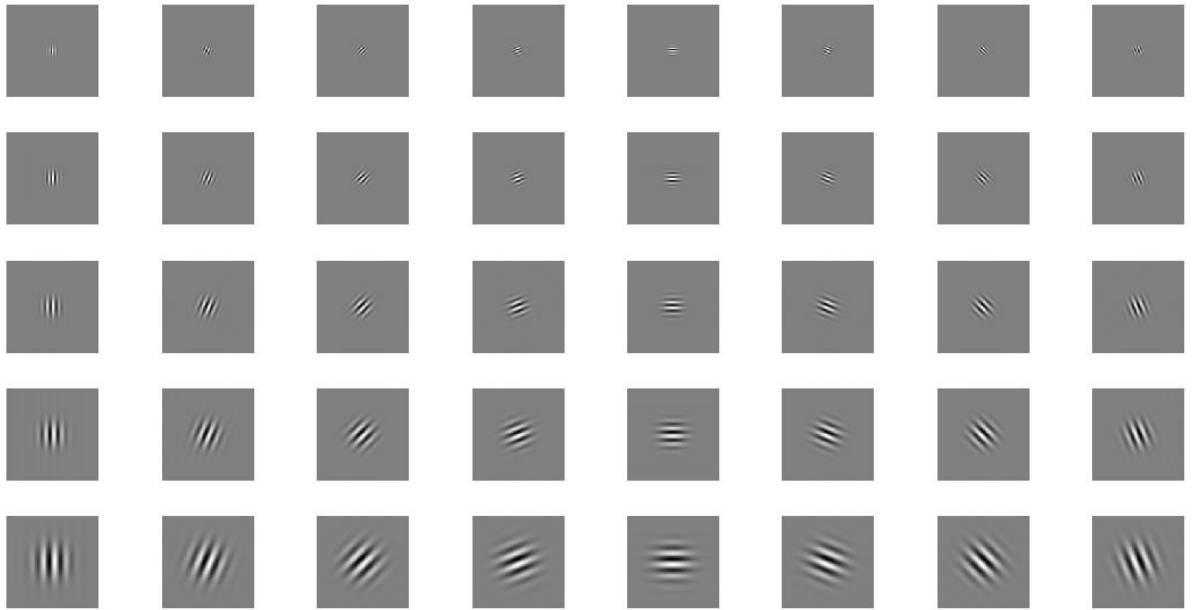


Figura 7.9. Wavelets de Gabor obtenidas con MATLAB para 5 frecuencias, 8 rotaciones. Área de 128x128 píxeles.

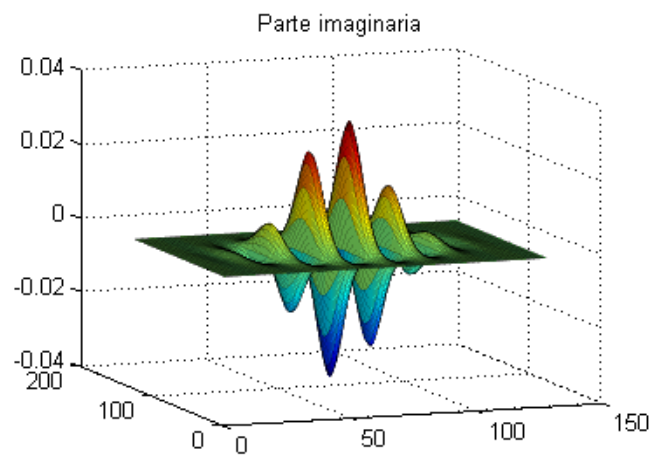
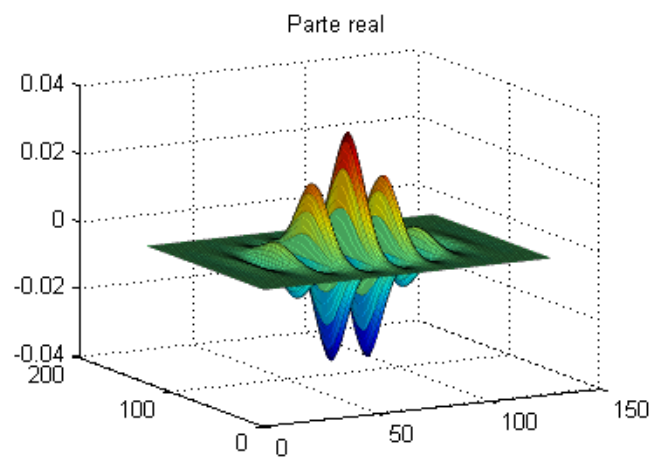


Figura 7.10. Representación tridimensional de una wavelet de Gabor obtenida con MATLAB.

Existen muchas consideraciones sobre el hecho de que las *wavelets* de Gabor no formen una base ortogonal, pero están todas relacionadas con la reconstrucción de la señal a partir de los coeficientes resultantes. En el entorno de este proyecto no tienen ninguna relevancia, ya que el objetivo de los coeficientes obtenidos no es utilizarlos para codificar y recuperar la imagen original.

Al convolucionar una imagen con una *wavelet* de Gabor, se observará una alta respuesta en las zonas de la imagen donde se correspondan la rotación de la *wavelet* con la dirección en la imagen de la componente frecuencial correspondiente a la frecuencia F_0 de la senoide del filtro (Ver Figura 7.11).

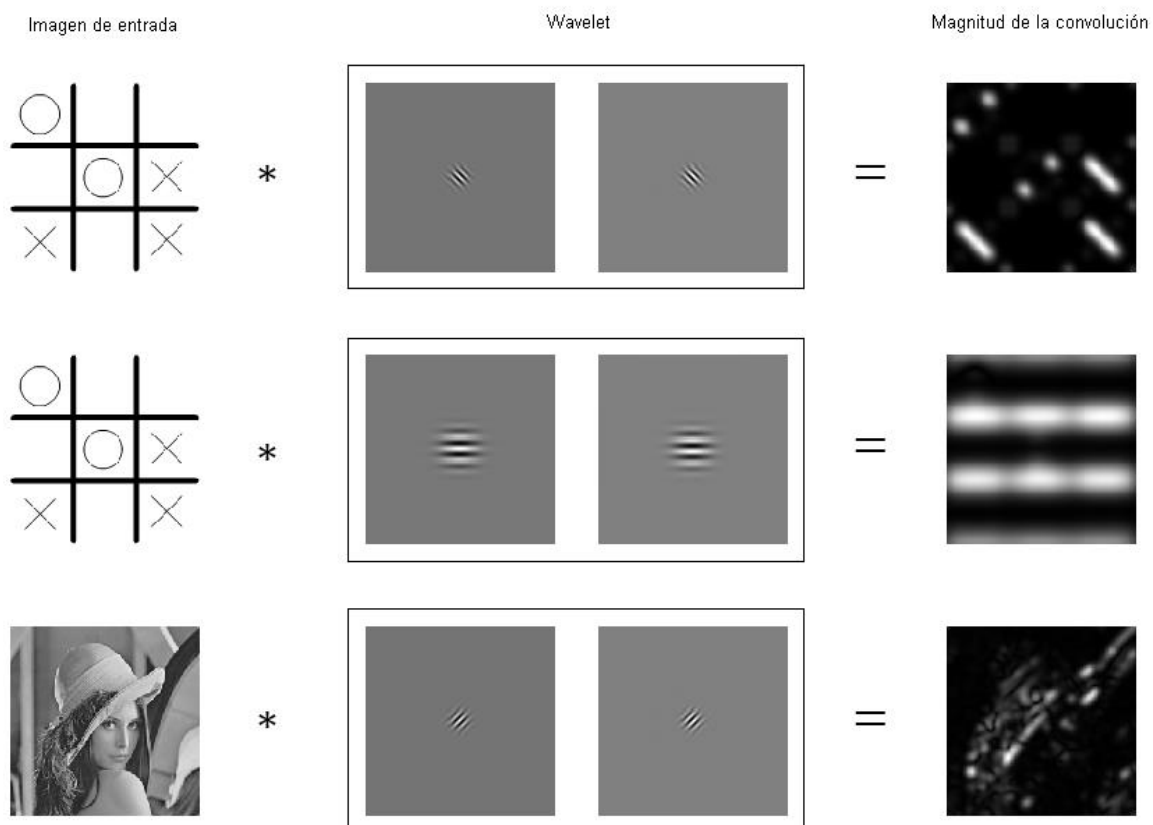


Figura 7.11. Resultado de convolucionar 3 wavelets distintas con imágenes. Tanto las wavelets, como la magnitud de las convoluciones fueron obtenidas en MATLAB.

7.4. Reconocimiento facial basado en *wavelets* de Gabor

Como es de suponer, el objetivo de las *wavelets* de Gabor en un sistema de reconocimiento facial es calcular los patrones de los individuos. Existen diferentes maneras de hacerlo, por ejemplo, una opción es convolucionar un conjunto de *wavelets* a determinadas frecuencias y rotaciones con las imágenes de entrenamiento de cada individuo. La magnitud del resultado de cada una de estas convoluciones podría ser una matriz de características, formando entre todas un patrón que describa al rostro de cada individuo. En la Figura 7.12 se observa la imagen facial de un individuo junto con sus

características calculadas mediante esta técnica propuesta. El conjunto de *wavelets* utilizado para calcular dichas características fue el de la Figura 7.9.



Figura 7.12. Características del individuo que se muestra en la fotografía, calculadas convolucionando la imagen con las wavelets de Gabor de la Figura 7.9. Estas gráficas, al igual que todas las anteriores, fueron obtenidas en MATLAB.

El inconveniente de utilizar este método es que para cada patrón se almacena una matriz completa del mismo tamaño de la imagen para cada característica calculada. Esto implica que para cada *wavelet* se almacenan n características, siendo n el número de píxeles de la imagen. Para el ejemplo de la Figura 7.12 el número de píxeles es $n = W \times H = 192 \times 256 = 49152$, que multiplicado por las 40 matrices de características, resulta en un total de 1.966.080 características a almacenar por cada patrón. Como consecuencia, en el momento de clasificar un individuo en el módulo de reconocimiento del sistema, deben compararse 1.966.080 características, lo que podría ralentizar el proceso de reconocimiento.

En vista del inconveniente antes comentado, la solución por la que se suele optar es convolucionar la imagen con las *wavelets* de igual manera, pero almacenar solamente el valor de dichas convoluciones para unos determinados píxeles. Esta solución es además

coherente porque al convolucionar una *wavelet* con una imagen, para píxeles vecinos el resultado es muy parecido; sobre todo a bajas frecuencias (evidentemente se perderá mucha información en altas frecuencias). En consecuencia, descartar las características cercanas a un píxel no supone una pérdida de información apreciable, sobre todo para bajas frecuencias. La pregunta que ahora surge es: ¿De cuáles píxeles se almacenan las características y de cuáles se descartan?

La respuesta vuelve a ofrecer un abanico de soluciones. Una de ellas por la que se suele optar es superponer una malla de puntos equiespaciados sobre la imagen (después de convolucionar) y almacenar las características de los píxeles que coincidan con dicha malla. En la Figura 7.13 se observa un ejemplo de una malla de 5x5 puntos sobre la imagen facial y una de las matrices de características de la Figura 7.12.

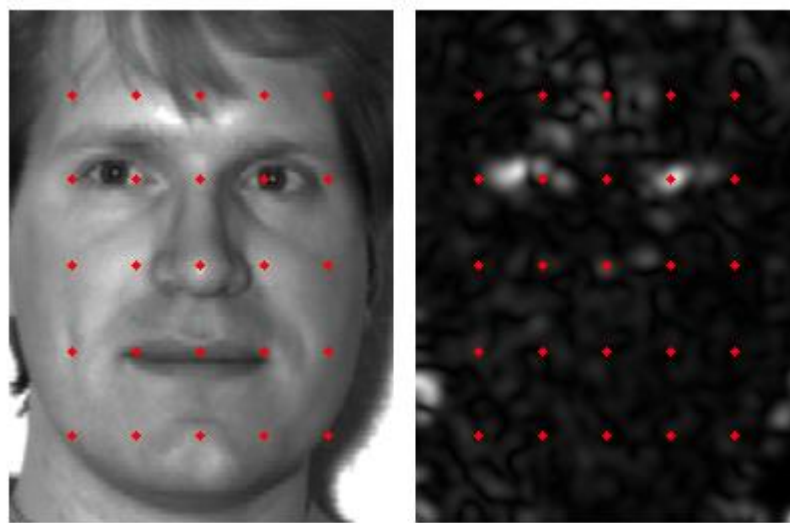


Figura 7.13. Ejemplo de una malla de 5x5 puntos sobre una imagen facial y las magnitudes del resultado de su convolución con una wavelet de Gabor. Las características a almacenar serán los valores de las matrices de características que coincidan con los puntos rojo.

Como resultado de este procedimiento, se reduce el número de características a comparar en el módulo de reconocimiento. Para la malla del ejemplo propuesto, el número de características por patrón se reduce hasta $5 \times 5 \times 40 = 1000$. Evidentemente, el número de puntos de la malla puede ser cualquiera que decida el diseñador del sistema de reconocimiento.

Otra opción podría ser que en lugar de ser una malla de puntos equiespaciados, sea una malla de puntos ubicados en puntos significativos como los ojos, la nariz o la boca. También se puede almacenar los coeficientes complejos resultantes de la convolución en lugar de almacenar su magnitud y compararlos por separado [33], o tomar también en cuenta la fase [34], etc. También hay autores que eligen la distancia entre los puntos de la malla de puntos basándose en la distancia que hay entre dos o más puntos significativos de la imagen [35].

Las características calculadas se suelen juntar en un vector para formar el patrón de una imagen. Al igual que con todos los sistemas de reconocimiento facial se almacenan

los patrones de un conjunto de entrenamiento en la base de datos y, cuando una imagen de entrada es presentada al sistema, se calcula su patrón y se compara con los almacenados para tomar una decisión.

Existen varias formas de comparar 2 patrones diferentes dependiendo del tipo de característica almacenado. Por ejemplo, si las características almacenadas fueron las magnitudes de las convoluciones, un método habitual suele ser calcular el producto escalar normalizado de ambos vectores de características y mientras más alto sea éste, más parecidas serán ambas imágenes.

Es conveniente recordar que la convolución, por definición, equivale a la proyección de una señal de entrada con una base, que es trasladada a lo largo de todo el tiempo. En el caso de la convolución bidimensional, esta definición no cambia, siendo exactamente lo mismo proyectar una imagen con una *wavelet* centrada en un píxel en específico, que tomar en dicho píxel el valor de la convolución de la misma imagen con la misma *wavelet*. Esto es lo mismo que decir que para un píxel $[x_0, y_0]$, se define la convolución (discreta) como:

$$f[x_0, y_0] * g[x_0, y_0] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1, n_2] g[x_0 - n_1, y_0 - n_2] \quad (7.20)$$

Para terminar, tampoco está de más recordar que debido a las características de la resolución en espacio y frecuencia de las *wavelets*, como resultado de proyectar una imagen sobre una familia de *wavelets* trasladadas a un píxel, el resultado de las bajas frecuencias contendrá información global de la imagen, mientras que el de las altas frecuencias contendrá información local de la imagen alrededor de ese píxel.

8. Emparejamiento por grafos elásticos (EGM)

El reconocimiento de objetos basado en emparejamiento por grafos elásticos fue introducido por primera vez por Lades et al. [36] como un sistema de reconocimiento invariante a la distorsión. Inicialmente, Lades et al. no pretendían que fuera un sistema especializado en reconocimiento facial aunque ha tenido mucha repercusión dentro de este ámbito. De hecho, a parte de las demostraciones sobre reconocimiento facial, llevaron a cabo demostraciones sobre clasificación de objetos de oficina.

Explicado en términos generales, el emparejamiento por grafos elásticos (EGM, por sus siglas en inglés) consiste en asignar un grafo a la imagen de un objeto, que puede verse superpuesto sobre la imagen. Cada vértice y cada arista de ese grafo estarán asociados con una etiqueta, que para los vértices dependerá de las características de la imagen alrededor del punto donde se encuentra dicho vértice, y para las aristas el vector distancia euclídea que hay entre dos vértices unidos por dicha arista. Cada vértice tiene además unas coordenadas o colocación \vec{x}_i en la imagen. A cada uno de los grafos asignados a una imagen se les llama **modelo** de dicha imagen. El término “elástico” se debe a que durante la etapa de clasificación, una copia deformable de cada modelo es asignada a la imagen de entrada y es distorsionada hasta reducir una función de coste.

La forma que adopta cada grafo depende del diseñador del sistema. Lades et al. diseñaron su sistema para clasificar objetos en imágenes de 128x128 donde los grafos elásticos eran rejillas de 7x10 vértices equiespaciados 11 píxeles entre ellos cuyas aristas los unían con sus vecinos horizontales y verticales. Otros autores hacen coincidir los vértices del grafo con puntos significativos del objeto de la imagen y conectarlos como mejor consideren. En el caso del reconocimiento facial, esto sería colocar los vértices del grafo en los ojos, los extremos de la boca, las cejas, la nariz, etc. [31] [34] [33]. En la Figura 8.1 se puede observar el grafo de Lades et al. superpuesto sobre la imagen de un individuo (izquierda) y otro de Bolme con vértices posicionados en puntos significativos de la imagen (derecha).

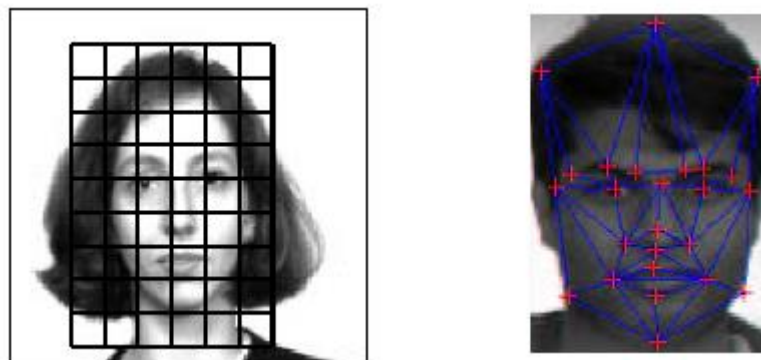


Figura 8.1. Grafos utilizados por Lades et al. [36] (a la izquierda) y Bolme [34] (a la derecha) en sus trabajos.

En casi la totalidad de los casos, las etiquetas de los vértices son un conjunto de características, resultado de proyectar la imagen sobre un conjunto de *wavelets* de Gabor

desplazadas al vértice. Sin embargo también es posible utilizar otras técnicas que midan las características de la imagen alrededor del vértice, como por ejemplo, los *Local Binary Patterns* [33]. Es importante destacar que para el EGM se utiliza solamente el módulo de los coeficientes complejos de Gabor, pero existen otros algoritmos basados en el EGM donde se utilizan los coeficientes complejos en sí (módulo y fase).

A un conjunto completo de características para cada vértice se le denomina **jet** (en el caso de que sean calculadas con *wavelets* de Gabor), por lo que se puede afirmar que las etiquetas de los vértices son los jets en cada uno de ellos. Cada jet contendrá tantas características como el número de *wavelets* de Gabor utilizadas, y cada modelo tendrá tantos jets como vértices tenga. Por ejemplo, si para calcular las características de Gabor se utilizaran las 40 *wavelets* de la Figura 7.9, entonces para un grafo como el de Lades et al. habría $70 \text{ vértices} \times 40 \frac{\text{características}}{\text{jet}} = 2800 \frac{\text{características}}{\text{modelo}}$. Se puede ver a los jets como vectores columna, donde cada fila es la magnitud del resultado de la proyección de una *wavelet* con la imagen o el coeficiente complejo en sí. En la Figura 8.2 se muestra una representación gráfica de cómo sería la asociación de un grafo a una imagen. El grafo fue generado en MATLAB con una forma muy sencilla para poder representar mejor las etiquetas.

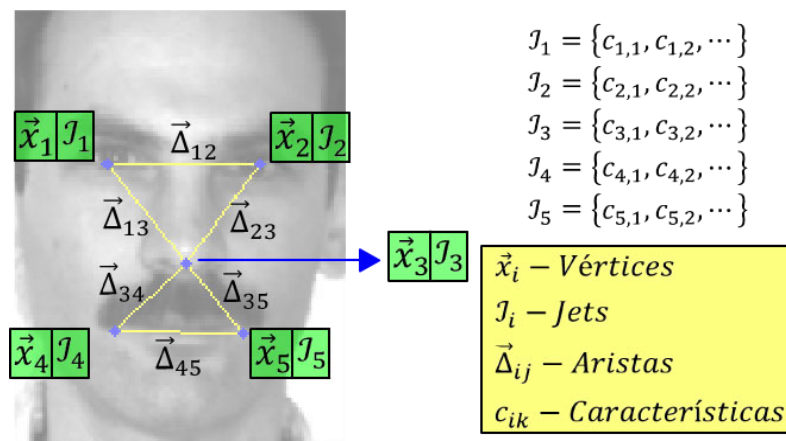


Figura 8.2. Asociación de un grafo a una imagen..

8.1. Etapa de entrenamiento

En la etapa de entrenamiento, se le asigna un modelo M a una imagen de cada individuo, que consistirá en un grafo cuyos vértices y aristas son identificados con sus respectivas etiquetas. La etiqueta de cada vértice será un jet, correspondiente a la proyección de un conjunto de *wavelets* de Gabor desplazadas al vértice, mientras que la etiqueta de cada arista será el vector distancia euclídea entre los dos vértices que une.

Una vez calculados los modelos de cada imagen de entrenamiento, se puede permitir descartar las imágenes, pues éstas quedan descritas por sus respectivos modelos.

8.2. Etapa de clasificación

En la etapa de clasificación, cada modelo M de la base de datos es comparado individualmente con la imagen de entrada, asignando una copia deformable I de su grafo a dicha imagen y recalculando las etiquetas de sus vértices. El resultado de dicho cálculo da lugar a un coste, que mide la diferencia entre el modelo y la imagen. Una vez calculado el coste, se va cambiando la colocación de los vértices de I siguiendo ciertas restricciones y se vuelve a medir el coste. Si este último coste es menor que el primero, se registra la nueva colocación como los vértices de I , y en caso contrario, se mantiene la colocación anterior. Este proceso se repite cuantas veces sea necesario para reducir el coste.

A continuación se muestran los cálculos necesarios para la comparación de un modelo M con un grafo deformable I :

Para comparar la similitud entre dos vértices con jets \mathcal{J}^I y \mathcal{J}^M , se utiliza su producto escalar normalizado, siendo más parecidos cuanto mayor sea su valor.

$$S_v(\mathcal{J}^I, \mathcal{J}^M) = \frac{\mathcal{J}^I \cdot \mathcal{J}^M}{\|\mathcal{J}^I\| \|\mathcal{J}^M\|} \quad (8.1)$$

Por otro lado, si se define el vector distancia euclídea entre dos vértices \vec{x}_i y \vec{x}_j como:

$$\vec{\Delta}_{ij} = \vec{x}_j - \vec{x}_i \quad (8.2)$$

para medir la similitud entre dos aristas $\vec{\Delta}_{ij}^I$ y $\vec{\Delta}_{ij}^M$, se utiliza la función de comparación cuadrática:

$$S_e(\vec{\Delta}_{ij}^I, \vec{\Delta}_{ij}^M) = (\vec{\Delta}_{ij}^I - \vec{\Delta}_{ij}^M)^2 \quad (8.3)$$

siendo más parecidas cuanto menor sea el valor de s_e .

Finalmente, el proceso de conseguir la deformación del grafo I que mejor se adapte al modelo M , es equivalente a conseguir una colocación de los vértices de I , $\{x_i^I\}$ tal que minimice la función de coste:

$$\begin{aligned} C_{total}(\{x_i^I\}) &= \lambda C_e + C_v \\ &= \lambda \sum_{(i,j) \in E} S_e(\vec{\Delta}_{ij}^I, \vec{\Delta}_{ij}^M) - \sum_{i \in V} S_v(\mathcal{J}^I(x_i^I), \mathcal{J}_i^M) \end{aligned} \quad (8.4)$$

Donde C_e y C_v son los costes debidos a todas las aristas y a todos los vértices, respectivamente, y λ es un coeficiente de rigidez que penaliza la distorsión del grafo I con respecto al grafo del modelo M .

Nótese que al principio I tiene la misma forma que M (es una copia), por lo tanto las etiquetas de las aristas de ambos grafos es idéntica y no generan coste C_e . Esto significa

que $C_e = 0$. El coste a minimizar es entonces la suma negativa de la similitud entre los jets de cada vértice. Supóngase que a este primer coste se le llama C_1 .

Si se supone que la primera deformación es cambiar la colocación del vértice \vec{x}_i , entonces cambiarán tanto los jets de \vec{x}_i (cambia C_v), como las etiquetas de las aristas que lo conectan con otros vértices (Cambia C_e , que ahora es distinto de cero), generando un nuevo coste C_2 . Si se supone que $\lambda = 1$, entonces la nueva posición de \vec{x}_i es registrada si $C_2 < C_1$.

Nótese que si se aumenta el valor de λ , también lo hace el valor de C_2 , con lo que si λ se hace demasiado grande ($\lambda \rightarrow \infty$) entonces no habrá ninguna colocación de los vértices de I que tenga un coste $C_2 < C_1$, y en consecuencia se mantendrá siempre la colocación inicial de los vértices. Esto es equivalente a tener un grafo rígido.

Para el extremo contrario, si $\lambda = 0$, la función de coste dependerá exclusivamente del coste causado por la similitud de los jets de todos los vértices. Esto sería un grafo extremadamente flexible.

El proceso de recolocación de un vértice se puede observar en la Figura 8.3.

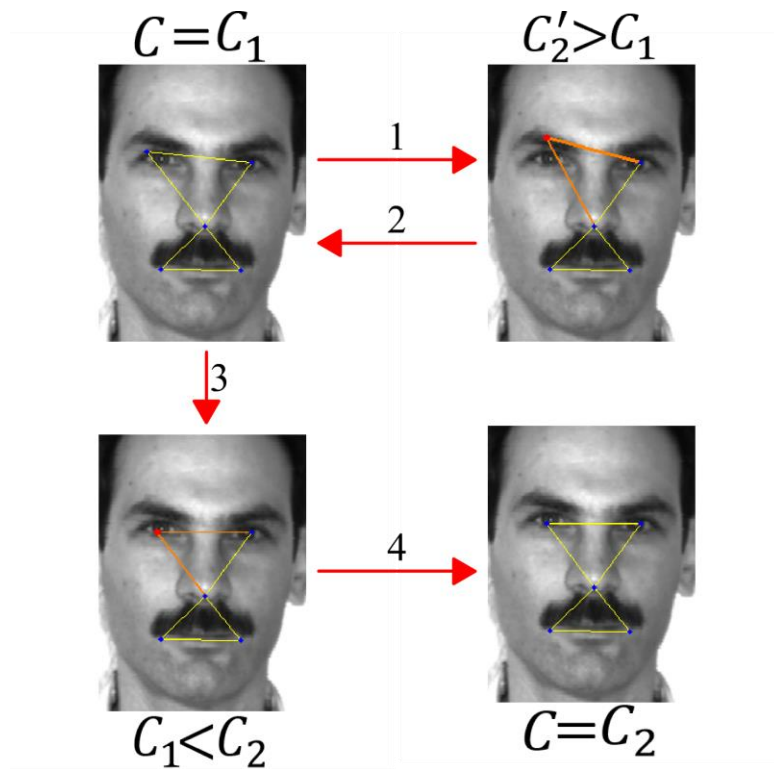


Figura 8.3. Recolocación de un vértice.

Sabiendo comparar un modelo con el grafo deformable de una imagen, es posible continuar con la etapa de clasificación en sí. Lades et al. utilizan un método de dos fases a las que el autor de este proyecto clasifica como **fase de colocación** y la **fase de elasticidad**.

Durante la fase de colocación, se coloca el grafo con una rigidez $\lambda \rightarrow \infty$ sobre la imagen en busca de la posición general del grafo que minimice el primer coste antes pasar a la etapa de elasticidad. Lades et al. colocan el grafo en el centro y lo hacen moverse con una holgura de 10 píxeles. Además, utilizan solamente las bajas frecuencias de las *wavelets* para llevarlo a cabo.

Durante la fase de elasticidad, se va modificando iterativamente la colocación de los vértices de I con una rigidez λ específica y si el coste $C_{k+1} < C_k$, se registra la nueva colocación de los vértices. En esta fase, Lades et al. van cambiando la colocación de los vértices de manera secuencial y en orden aleatorio. Cada vez que se accede a un vértice, éste se recoloca en una dirección aleatoria y una distancia aleatoria hasta una distancia máxima de la mitad de la longitud original de las aristas que lo conectaban con otros vértices. Si una nueva colocación reduce el coste, se registraba y se reiniciaba el proceso (se igualaba el número de iteraciones a cero), y en caso contrario se repetía la recolocación aleatoria. Se abandonaba la recolocación de dicho vértice para pasar al siguiente cuando se haya alcanzado un número específico de iteraciones sin éxito de disminuir el coste.

Una vez terminada la fase de elasticidad, se considera el último coste como el coste mínimo y se dirá que el grafo I es lo más parecido posible a M . El proceso debe repetirse con todos los modelos de la base de datos, y el modelo con el cual I haya producido el menor coste es elegido como resultado de la etapa de clasificación.

8.3. Elastic Bunch Graph Matching

El *Elastic Bunch Graph Matching* (EBGM) es una técnica de reconocimiento introducida por Wiskott et al. [31], y que al igual que el EGM no fue diseñada para sistemas especializados en reconocimiento facial. La principal diferencia que presenta frente al EGM es que genera un *Face Bunch Graph* (FBG), que es un grafo cuya forma es la forma media de todos los modelos de entrenamiento, pero la etiqueta de sus vértices son todos los jets de los modelos de entrenamiento para cada vértice. La ventaja de utilizar un FBG es que puede ser utilizado para generar los modelos de entrada automáticamente, aunque Wiskott et al. indican que empiezan a resultar modelos fiables una vez generados manualmente unos 70 modelos.

El motivo de que no sea utilizado el EBGM en este proyecto es que no aportada ninguna información adicional relevante a parte del FBG.

PARTE III: IMPLEMENTACIÓN

9. Descripción general de la implementación

Como resultado de la investigación realizada, se llevó a cabo la implementación de un programa de reconocimiento facial formado por cuatro sistemas de reconocimiento que funcionan tanto en modo verificación, como en modo identificación. A su vez, el modo identificación puede trabajar en conjunto cerrado o abierto dependiendo de los parámetros con los que el usuario configure al sistema. Adicionalmente, se propone un modo híbrido de verificación que utiliza los resultados del modo identificación para llevar a cabo la verificación de una identidad con el objetivo de reducir las tasas de falsa aceptación.

Cada uno de los sistemas que forman parte del programa es independiente de los demás y utiliza una de las cuatro técnicas de reconocimiento estudiadas en el marco teórico (PCA, LDA, *wavelets* de Gabor y EGM). Además, se ofrece al usuario la posibilidad de modificar algunos parámetros de funcionamiento de todos los sistemas para poder compararlas y elegir la que mejor se adapte a sus necesidades.

El programa cuenta también con la funcionalidad de crear y modificar una propia base de datos con etiquetas (enlace de cada imagen con la identidad de un individuo) para que cada sistema de reconocimiento facial implementado pueda valerse de dicha base de datos.

Adicionalmente al programa de reconocimiento facial, se llevó también a cabo la implementación de un programa de evaluación que permite crear un set de validación de rostros de los individuos incluidos en la base de datos y generar para cada sistema las curvas de evaluación estudiadas en el marco teórico para hacer posible su comparación de una manera objetiva (2.3. Evaluación de los modos de funcionamiento).

El sistema está dotado además de otras características que incrementan su utilidad y lo hacen apto para un mayor tipo de situaciones. Entre ellas destacan: Almacenamiento de un historial de clasificación, herramienta de detección y segmentación facial con porcentaje de corte adicional, posibilidad de elegir un formato de imágenes, compatibilidad con cámara web, visualización de *eigenfaces* y *fisherfaces* resultantes de los entrenamientos para PCA y LDA, herramienta de creación de grafos a partir de una imagen de ejemplo y herramienta de creación de modelos a partir de un grafo establecido anteriormente, entre otras más.

La utilidad que se pretende dar al programa es que sirva como una base de diseño para una aplicación real, donde la decisión del sistema sea consecuencia de otras acciones para las cuales se solicite una clasificación (acceso a una información o lugar, identificación criminal, etc.).

Por ejemplo, suponiendo que una empresa deseara automatizar el acceso a través de un sistema de reconocimiento facial, con esta implementación podría crear una base de datos con las caras de sus empleados y utilizar el modo verificación de cualquiera de las técnicas estudiadas (excepto EGM, que como se explicará más adelante, solo funciona en modo identificación).

Otro ejemplo sería un sistema de etiquetado de fotos como el que se presenta en algunas redes sociales. Supóngase que un individuo desea hacer un álbum de fotos virtual de su boda donde además se muestren las etiquetas de las personas que se encuentran en cada foto para no olvidar los nombres de la familia de su pareja. En este caso se podría utilizar cualquiera de los sistemas en modo identificación, y con unas pocas modificaciones sobre el código fuente del programa (que serán comentadas más adelante), sería posible crear un programa que consiga dicho objetivo con poco esfuerzo.

Otra posibilidad es utilizar el programa con fines educativos, de tal manera que tanto el uso del programa en sí, como el código en MATLAB del mismo pueden servir como material docente o de investigación para el estudio del reconocimiento facial.

En la Figura 9.1 se presenta el diagrama de bloques general de los módulos de reconocimiento implementados.

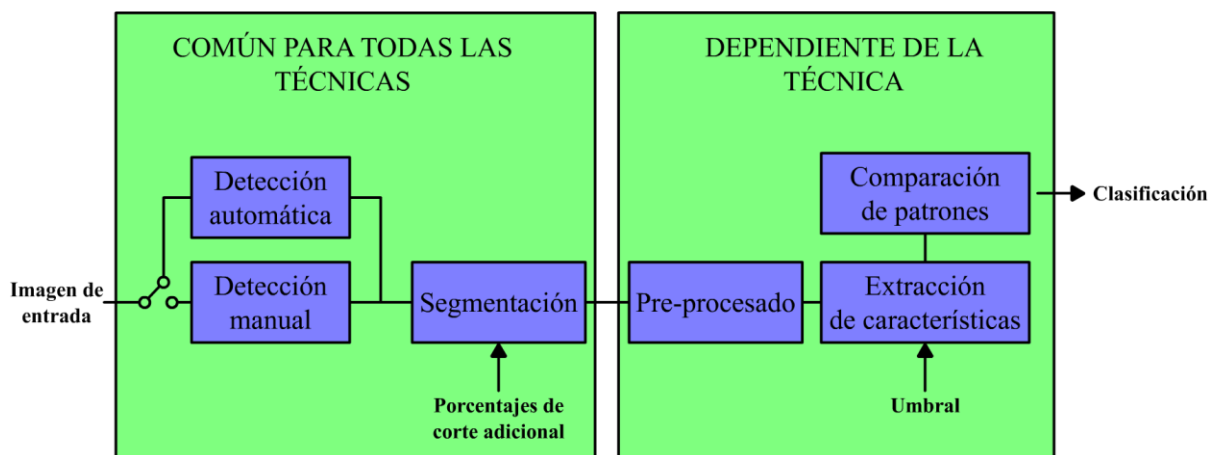


Figura 9.1. Diagrama de bloques general de los módulos de reconocimiento implementados.

9.1. Detección automática

Para llevar a cabo la detección automática se utilizó el objeto de MATLAB *CascadeObjectDetector*, que utiliza un clasificador entrenado por defecto aunque permite la posibilidad de entrenar otro clasificador distinto. El objeto *CascadeObjectDetector* utiliza el algoritmo de Viola-Jones y se encuentra en el *toolbox: Computer Vision System Toolbox*.

La implementación llevada cabo ofrece la posibilidad de hacer una selección manual de la zona de la cara en el caso de que el detector falle, o no haya sido lo suficientemente preciso; o que la versión de MATLAB que se esté usando no incluya el *toolbox* antes mencionado.

9.2. Estructura de diseño

Aunque en los anexos de este documento se dispone de un manual de uso del programa, se considera preferible conocer el esquema de navegación entre las diferentes ventanas del programa antes de explicar la estructura de diseño. De esta manera se da una idea general de las diferentes posibilidades que se ofrecen y cómo acceder a ellas. Dicho esquema de navegación se puede observar en la Figura 9.2.

Es importante destacar que el programa cuenta con otras ventanas secundarias que no aparecen en el esquema por simplicidad. Éstas son: La ventana de adquisición desde cámara web, la herramienta de detección y segmentación de rostros, la de etiquetado de imágenes y la ventana de clasificación fallida.

Una vez presentado el esquema de navegación, en la Figura 9.3 se muestra la estructura de diseño del programa realizado. En ella se observan los directorios y ficheros fundamentales para el funcionamiento del programa.

Menú principal

- Menú de análisis por componentes principales (PCA)
 - Ventana de clasificación en modo identificación
 - Ventana de clasificación en modo verificación
 - Ventana de propiedades de la clasificación por PCA
 - Ventana de visualización de media y *eigenfaces*
- Menú de análisis lineal discriminante (LDA)
 - Ventana de clasificación en modo identificación
 - Ventana de clasificación en modo verificación
 - Ventana de propiedades de la clasificación por LDA
 - Ventana de visualización de media y *fisherfaces*
- Menú de *wavelets* de Gabor
 - Ventana de clasificación en modo identificación
 - Ventana de clasificación en modo verificación
 - Ventana de propiedades de la clasificación por *wavelets* de Gabor
 - Ventana de visualización de las *wavelets* utilizadas y malla de puntos
- Menú de emparejamiento de grafos elásticos (EGM)
 - Ventana de clasificación en modo identificación
 - Gestor de modelos
 - Herramienta de creación de grafos
 - Herramienta de creación de modelos
 - Ventana de propiedades de la clasificación por EGM
 - Ventana de visualización de las *wavelets* utilizadas
- Menú de la base de datos
 - Ventana de inscripción de individuos
 - Ventana de edición de individuos
 - Ventana de eliminación de individuos
 - Ventana de actualización desde historial
- Ventana de propiedades generales

Figura 9.2. Esquema de navegación del programa de reconocimiento facial.

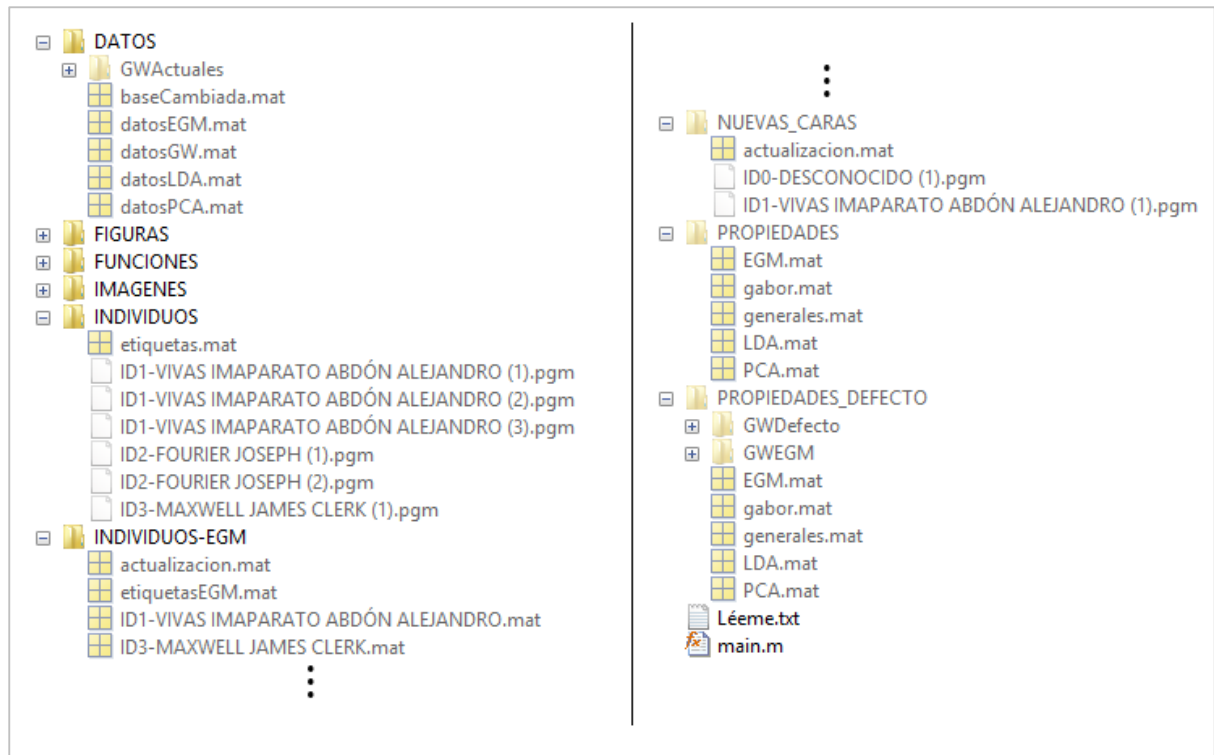


Figura 9.3. Estructura de diseño del programa.

Los directorios que en la figura se observan son los siguientes:

- **DATOS:** Contiene los datos resultantes del entrenamiento de cada uno de los sistemas y un archivo de control (“baseCambiada.mat”) formado por variables de tipo “flag” que indican cuáles de los sistemas necesitan ser reentrenados o actualizados debido a modificaciones en la base de datos.
- **FIGURAS:** Contiene el código fuente de todas las ventanas que forman parte de la interfaz gráfica del programa, mientras que el directorio “FUNCIONES” contiene las funciones a las que accede la interfaz gráfica para llevar a cabo las peticiones del usuario.
- **IMAGENES:** Contiene las imágenes que son mostradas en algunas ventanas de la interfaz gráfica.
- **INDIVIDUOS:** Contiene las imágenes de cada individuo incluidas en la base de datos y el archivo “etiquetas.mat” donde se encuentra la matriz de etiquetas que asigna a cada imagen una identidad. Más adelante se explicará cómo se gestiona la base de datos para que cada sistema de reconocimiento pueda acceder a la misma.
- **INDIVIDUOS-EGM:** Es un directorio donde se almacena exclusivamente información relativa al sistema de reconocimiento por emparejamiento de grafos elásticos. Su contenido será descrito más adelante.
- **NUEVAS_CARAS:** En él se almacena el historial de clasificación.
- **PROPIEDADES y PROPIEDADES_DEFECTO:** Como sus nombres indican, contienen las **propiedades modificables** de cada uno de los sistemas y las propiedades por defecto de los mismos, respectivamente.

Obsérvese además que el único código fuente que se encuentra en el directorio raíz del programa es la función *main*. Más adelante se incluirá también el programa de evaluación a la estructura.

9.3. Gestión de la base de datos y propiedades generales

El programa implementado permite ingresar imágenes a la base de datos a partir de archivos o por cámara web. Antes de añadir cualquier imagen, éstas deben ser etiquetadas por el usuario con la identidad de un individuo (con la ayuda de la ventana de etiquetado), registrando así dicho individuo en la base de datos. Una vez que un individuo es registrado, se pueden enlazar más imágenes a su identidad, por lo que las imágenes son almacenadas en el directorio “INDIVIDUOS” con nombre:

$$\text{ID}[X]\text{--[APELLIDOS] [NOMBRE] ([NÚMERO DE IMAGEN])}.EXT$$

Donde X es un número que identifica la posición del individuo en la base de datos y EXT es la extensión de imagen especificada en las propiedades generales.

Adicionalmente cada individuo se registra en un archivo llamado “etiquetas.mat” que está formado por una matriz donde cada fila es la etiqueta correspondiente a cada individuo. La matriz de etiquetas tiene 3 columnas y tantas filas como individuos tenga la base de datos. El número de identificación de cada individuo corresponde con su posición en la matriz de etiquetas. El formato de dicha matriz se presenta a continuación:

$$etiquetas = \begin{bmatrix} \text{etiqueta 1} \\ \text{etiqueta 2} \\ \vdots \\ \text{etiqueta N} \end{bmatrix}$$

$$= \begin{bmatrix} ID1 & [APELLIDOS NOMBRE] & [NÚMERO TOTAL DE IMÁGENES] \\ ID2 & [APELLIDOS NOMBRE] & [NÚMERO TOTAL DE IMÁGENES] \\ \vdots & \vdots & \vdots \\ IDN & [APELLIDOS NOMBRE] & [NÚMERO TOTAL DE IMÁGENES] \end{bmatrix}$$

Nótese que como cada individuo tiene un número de identificación, puede darse la posibilidad de que dos individuos compartan el mismo nombre y apellidos, diferenciándose tan solo en su número de identificación e imágenes enlazadas a ellos en la base de datos.

En vista de que se considera toda la base de datos como el conjunto de entrenamiento del programa, cada vez que un individuo es añadido, eliminado, o editado, Se activan los *flags* contenidos en el fichero “baseCambiada” para los sistemas de reconocimiento por PCA, LDA y *wavelets* de Gabor. Para el sistema de reconocimiento por EGM solamente se activa el *flag* cuando un individuo es editado o eliminado de la base

de datos debido a que como se explicará más adelante, éstos son los únicos cambios que le afectan.

Cuando se elimina un individuo, todos aquellos con números de identificación más altos ajustan su identificación a su nueva posición en la matriz de etiquetas. Este cambio se ve también reflejado en los nombres de las imágenes almacenadas. De igual manera, cuando se edita el nombre de alguno o se borra una de sus imágenes, los cambios se ven reflejados tanto en la matriz de etiquetas, como en las imágenes almacenadas.

Otra posibilidad que ofrece el sistema es la modificación de ciertas propiedades generales, tales como seleccionar la extensión y dimensión que tendrán las imágenes de la base de datos (Solo modificables antes de añadir al primer individuo en la base de datos), seleccionar el dispositivo de captura, cambiar la resolución de captura, y elegir un porcentaje de corte adicional para la herramienta de detección y segmentación de rostros.

De entre todas las propiedades generales existe una cuyo uso no parece muy evidente: El porcentaje de corte adicional para la herramienta de detección y segmentación. El objetivo de esta propiedad es ofrecer al usuario una ayuda para aumentar la rapidez del proceso de detección

En la Figura 9.4 se muestran dos ejemplos de detección y segmentación para la misma imagen. La primera de ellas (Figura 9.4.a) representa una selección manual en la que se decidió segmentar la zona limitada por el rectángulo que forman las cejas y barbilla, y la segunda (Figura 9.4.b) representa una detección automática.

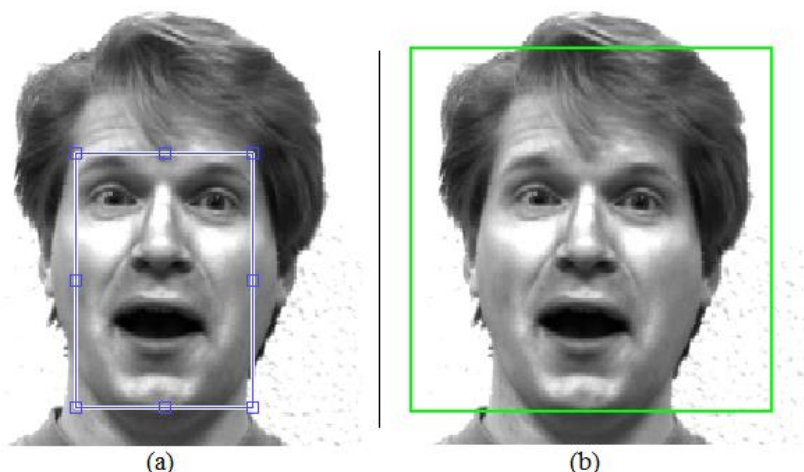


Figura 9.4. Ejemplos de detección y segmentación para una misma imagen.

Supóngase que el usuario deseara un corte como el de la Figura 9.4.b, pero no dispone del *toolbox* de MATLAB que permite la detección automática. Con el fin de conseguir un corte más uniforme en todas las imágenes, el usuario puede configurar un porcentaje de corte adicional de tal manera que seleccionando una zona más sencilla de remarcar a primera vista como la de la Figura 9.4.a, la herramienta de detección y segmentación entregue a su salida un corte como el de la Figura 9.4.b.

Supóngase ahora el caso contrario: El usuario desea un corte como el de la Figura 9.4.a, pero la detección automática ofrece uno como el de la Figura 9.4.b. En caso de no existir una configuración de corte adicional, el usuario tendrá que desechar constantemente el resultado de la detección automática para hacer una selección manual (haciendo obsoleta al detección automática). Por otro lado, si se configurase un corte adicional negativo, la herramienta de detección y segmentación entregará a su salida un corte como el de la Figura 9.4.a para una detección automática como la de la Figura 9.4.b.

Finalmente, se comenta que las propiedades generales pueden ser restauradas a unos valores por defecto mostrados en la Tabla 9.1:

Propiedades de cámara			
Adaptador	<i>winvideo</i> , si existiese. En caso contrario, se deshabilita la captura por cámara web hasta que el usuario la configura en la ventana de propiedades generales.		
Dispositivo	El primero que se detecte conectado al adaptador <i>winvideo</i> . En caso de no haber ninguno, se deshabilita la captura por cámara web.		
Formato	El primero que se detecte que permita el dispositivo por defecto.		
Propiedades de cámara		Propiedades de cámara	
Porcentaje lateral	15	Ancho	192
Porcentaje superior	30	Alto	256
Porcentaje inferior	5	Formato	PGM

Tabla 9.1. Propiedades generales por defecto.

Para evitar el problema de la desconexión de dispositivos entre una ejecución del programa y otra, se optó por hacer que las propiedades de la cámara se restauren a las propiedades por defecto cada vez que se ejecuta el programa.

El porcentaje adicional por defecto fue escogido de tal manera que para una selección manual como la de la Figura 9.4.a, el corte abarque un poco más allá de los límites de la cabeza sin contar el pelo (sólo hasta la frente).

De forma acorde con el porcentaje adicional por defecto, el ancho y el alto de las imágenes a almacenar en la base de datos guarda una relación de aspecto de 0.75.

9.4. Historial de clasificación

El programa implementado ofrece la posibilidad de ampliar la base de datos a partir de un historial generado con las imágenes que han sido clasificadas ya sea de manera correcta, o de manera errónea para las clasificaciones realizadas en modo identificación.

Cada vez que se pone a prueba una técnica de clasificación, el usuario puede indicarle al programa si su tarea de clasificación fue exitosa o no. En el caso de que el sistema se haya equivocado, el usuario debe indicar la verdadera identidad del individuo. Consecuentemente, la imagen con la que se puso a prueba el sistema de clasificación es almacenada en un historial de clasificación (directorio “NUEVAS_CARAS” de la Figura 9.3) etiquetada con la identidad correcta.

Adicionalmente, en el directorio “NUEVAS_CARAS” se almacena también un fichero llamado “actualizacion.mat”. Dicho fichero está formado por una matriz de actualización de 4 columnas donde cada fila representa una operación de edición o exclusión de un individuo de la base de datos. En la Tabla 9.2 se observan los valores de cada columna de la matriz de actualización, que dependen la operación realizada por el usuario.

Operación	Columna 1	Columna 2	Columna 3	Columna 4
Cambio de nombre	[ID]	[APELLIDOS NOMBRE]	0	[APELLIDOS NOMBRE] Nuevos
Eliminación de una foto			Número de la foto que fue borrada	0
Eliminación de un individuo			-1	0

Tabla 9.2. Valores de cada columna de la matriz de actualización en función de la operación realizada.

La función de la matriz de actualización es guiar al programa para que los cambios realizados sobre la base de datos afecten de la misma forma al historial de clasificación y llevar a cabo una ampliación coherente y sin errores.

Un problema que puede surgir es que el usuario haya indicado al programa una identidad errónea durante la clasificación. Para solventarlo, una ventana de comprobación pide al usuario que confirme la identidad de todos los individuos almacenados en el historial de clasificación.

EL hecho de ampliar la base de datos con las muestras almacenadas en el historial de clasificación tiene efectos positivos para las próximas clasificaciones, puesto que de esta forma el sistema estará mejor entrenado.

Por ejemplo: supóngase que se desea llevar a cabo la implementación del álbum de fotos virtual que se comentó al principio del capítulo. Los primeros rostros serían etiquetados de forma manual, formando un conjunto de entrenamiento, mientras que los demás serían etiquetados de forma automática a partir del historial de clasificación.

9.5. Tipo de Umbral y discriminación del sistema ante individuos externos a la base de datos

Tanto para la implementación del sistemas de reconocimiento basado en análisis por componentes principales, como para el basado en análisis lineal discriminante se utilizan dos tipos de umbrales:

1. Umbral por medias: Para cada individuo se calculan las coordenadas medias, y la distancia que hay entre dicha media y la muestra más alejada de la misma. Un patrón de entrada es clasificado como legítimo de un individuo en el caso de que su distancia a la media de dicho individuo esté dentro de la distancia máxima calculada.

2. Umbral por muestras: Se guardan las coordenadas de todas las muestras, y para cada muestra la distancia mínima que hay hacia las otras muestras. Un patrón de entrada es clasificado como legítimo de un individuo en el caso de que su distancia a cualquiera de las muestras de dicho individuo esté dentro de la distancia mínima calculada para dicha muestra.

Ambos tipos de umbral pueden observarse en la Figura 9.5. Nótese que dependiendo del tipo de umbral se obtiene un resultado distinto para la muestra de ejemplo.

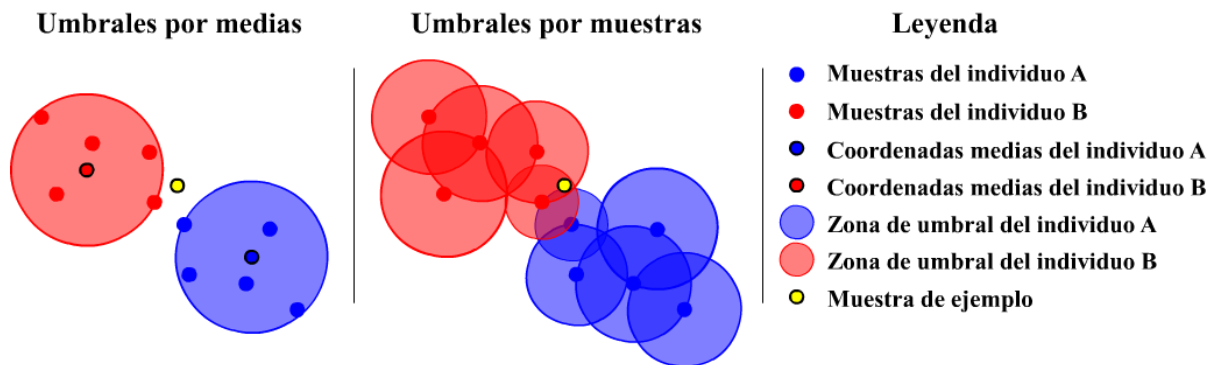


Figura 9.5. Tipos de umbral utilizados.

Para ofrecer al usuario la posibilidad de cambiar el umbral medido sin la necesidad de manejar números distintos cada vez que se entrena al sistema, se define un **factor de discriminación externa** que dilate o contraiga el umbral calculado. Una muestra formará parte de una clase (individuo) si su distancia d a la media de dicha clase, o a una muestra de la misma (dependiendo del tipo de umbral utilizado), se encuentra dentro de la zona limitada por el umbral original u_0 (umbral calculado durante el entrenamiento) multiplicado por un factor de discriminación externa α .

$$d < \alpha u_0 \rightarrow \frac{d}{u_0} < \alpha \quad (9.1)$$

Por lo tanto, e independientemente del sistema utilizado, el usuario podrá modificar el factor de discriminación externa, donde 0 es un valor completamente discriminativo (solamente sería admitido un patrón de entrada exacto al patrón almacenado) e infinito un valor nada discriminativo (todos los patrones de entrada serían admitidos).

Es evidente que durante el entrenamiento del sistema se obtendrán tantos umbrales originales como individuos haya en el caso del tipo de umbral por medias, y tantos umbrales originales como individuos haya en el caso del tipo de umbral por muestras. Si se optara por almacenar todos los umbrales originales distintos, podría surgir el problema de que un patrón de entrada se encuentre más cercano al patrón almacenado correcto que de otro patrón incorrecto, pero sin estar dentro de los límites del patrón correcto y sí dentro de los límites del patrón incorrecto (Figura 9.6, arriba). Para solventar este problema se optó por utilizar un umbral original único correspondiente con el umbral original más alto calculado durante el entrenamiento (Figura 9.6, abajo).

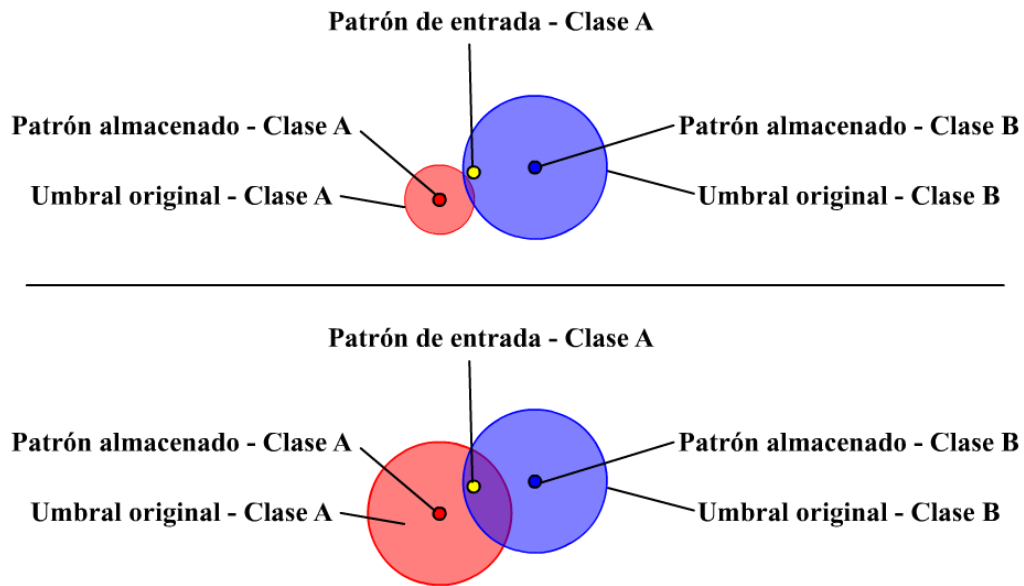


Figura 9.6. Problema que surge cuando se establecen varios umbrales originales (arriba) y solución estableciendo un umbral original único (abajo).

9.6. Sistema de reconocimiento facial basado en PCA

El diagrama de bloques correspondiente a la implementación del sistema de reconocimiento facial basado en PCA se muestra en la Figura 9.7.

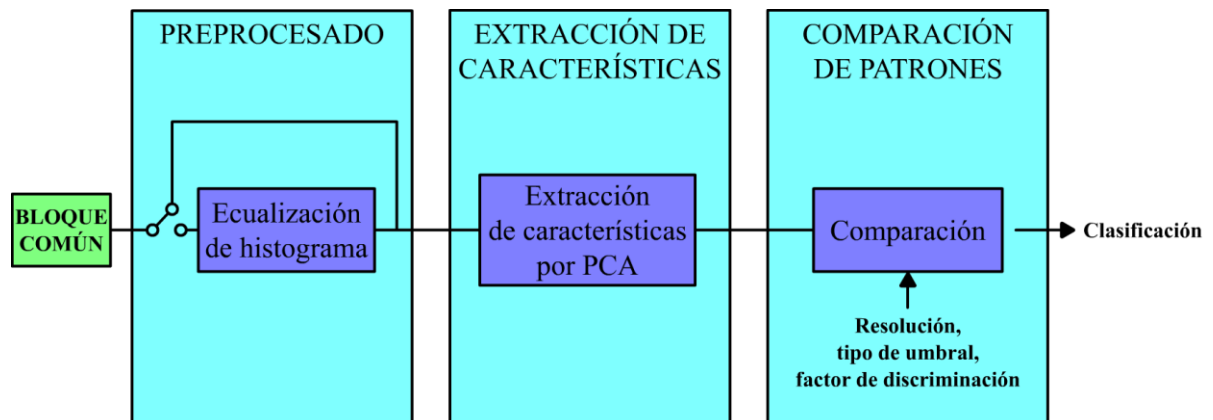


Figura 9.7. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en PCA. El bloque común es el mostrado en la Figura 9.1.

Propiedades

A cada una de ellas le corresponde una variable de las almacenadas en el archivo “PCA.mat” del directorio “PROPIEDADES” (Figura 9.3) y son las siguientes:

- Pre-procesado: Es el pre-procesado que será aplicado a las imágenes para el próximo entrenamiento del sistema (y consecuentemente para las posteriores clasificaciones). Éste puede ser escogido entre ecualización de histograma o ninguno.
- Tipo de umbral: tipo de umbral a calcular en el próximo entrenamiento (Umbral por medias o umbral por muestras).

- Factor de discriminación externa
- Resolución: Indica el número de *eigenfaces* a utilizar para la próxima clasificación. Reducir el número de *eigenfaces* a utilizar equivale a descartar aquellas con los autovalores más bajos.
- Resolución máxima: Número de *eigenfaces* calculadas en el último entrenamiento. Esta es la única propiedad que no está sujeta a modificaciones del usuario.

Es importante comentar que el usuario nunca modifica directamente el número de *eigenfaces* a utilizar, sino que en lugar de ello modifica el **porcentaje de varianza mínimo que desea representar** (ver capítulo 5), como se puede observar en la Figura 9.8. La interfaz del programa se encarga de calcular el número de *eigenfaces* correspondientes para almacenarlo en la variable de resolución. Posteriormente, el programa actualiza la resolución a la resolución real (el porcentaje de información exacto representado por las *eigenfaces* utilizadas) y muestra al usuario el menor número entero más cercano a dicha resolución.

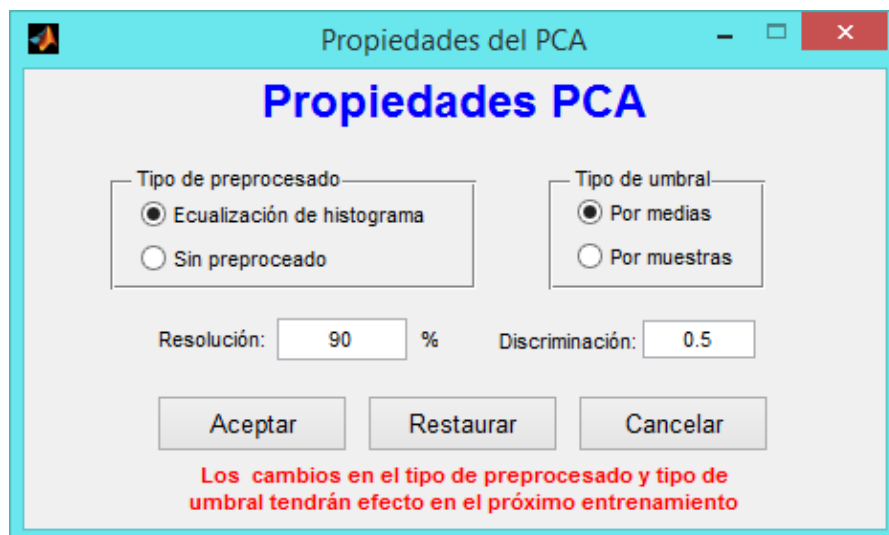


Figura 9.8. Ventana de propiedades del sistema de reconocimiento basado en PCA.

Luego de que el usuario haya modificado las propiedades, el sistema ofrece la posibilidad de volver a recuperar unas propiedades por defecto (Tabla 9.3).

Propiedades por defecto del sistema de reconocimiento basado en PCA	
Pre-procesado	Ecualización de histograma
Tipo de umbral	Por medias
Factor de discriminación	0,5
Resolución	Aquella que asegure una representación del 90% de varianza como mínimo para el último entrenamiento
Resolución máxima	Número de <i>eigenfaces</i> resultantes del último entrenamiento

Tabla 9.3. Propiedades por defecto del sistema de reconocimiento facial basado en PCA.

Datos

Se refieren a los datos que se almacenan al finalizar un entrenamiento del sistema de reconocimiento facial basado en PCA y están contenidos en el fichero “datosPCA.mat” del directorio “DATOS” (Figura 9.3). Son los siguientes:

- Autovalores, *eigenfaces*, cara media y varianza total de los datos de entrada.
- Tipo de pre-procesado utilizado durante el entrenamiento y tipo de umbral calculado. Es evidente que para llevar a cabo la clasificación, el tipo de pre-procesado aplicado a la entrada debe ser el mismo que el aplicado a cada imagen de entrenamiento.
- El resto de los datos dependen del tipo de umbral calculado. Para el umbral por medias se guardan las coordenadas medias de cada individuo (patrón medio) y sus respectivos umbrales, y para el umbral por muestras se guardan las coordenadas de todas las muestras de la base de datos y sus respectivos umbrales.

Cabe destacar que en realidad la resolución máxima no es una propiedad, sino un dato. No obstante, es considerada como una propiedad para no tener ambas variables correspondientes almacenadas en distintos directorios.

9.7. Sistema de reconocimiento facial basado en LDA

El diagrama de bloques correspondiente a la implementación del sistema de reconocimiento facial basado en PCA se muestra en la Figura 9.9.

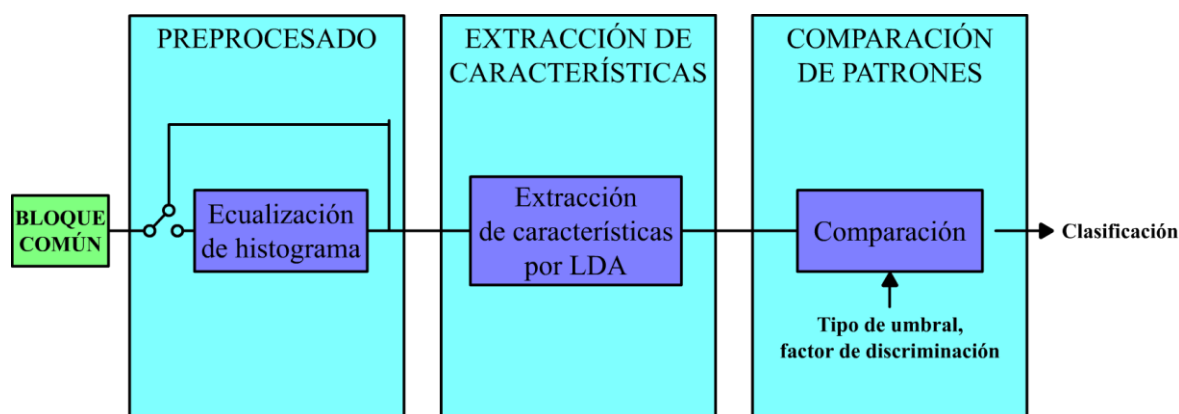


Figura 9.9. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en LDA. El bloque común es el mostrado en la Figura 9.1.

Propiedades

A cada una de ellas le corresponde una variable de las almacenadas en el archivo “LDA.mat” del directorio “PROPIEDADES” (Figura 9.3) y son las siguientes:

- Pre-procesado: Es el pre-procesado que será aplicado a las imágenes para el próximo entrenamiento del sistema (y consecuentemente para las posteriores

clasificaciones). Éste puede ser escogido entre ecualización de histograma o ninguno.

- Tipo de umbral: tipo de umbral a calcular en el próximo entrenamiento (Umbral por medias o umbral por muestras).
- Factor de discriminación externa

Los valores por defecto se observan en la Tabla 9.4.

Propiedades por defecto del sistema de reconocimiento basado en LDA	
Pre-procesado	Ecualización de histograma
Tipo de umbral	Por medias
Factor de discriminación	6

Tabla 9.4. Propiedades por defecto del sistema de reconocimiento facial basado en LDA.

Datos

Se refieren a los datos que se almacenan al finalizar un entrenamiento del sistema de reconocimiento facial basado en LDA y están contenidos en el fichero “datosLDA.mat” del directorio “DATOS” (Figura 9.3). Son los siguientes:

- Cara media, autovectores resultantes de la reducción de dimensión por PCA y autovectores resultantes del LDA (6. Análisis lineal discriminante (LDA)).
- Tipo de pre-procesado utilizado durante el entrenamiento y tipo de umbral calculado.
- Resto de datos dependientes del tipo de umbral calculado. Mismos que para PCA.

9.8. Sistema de reconocimiento facial basado en *wavelets* de Gabor

Cuando se diseña un sistema de reconocimiento basado en *wavelets* de Gabor hay que tomar una decisión de compromiso entre tiempo de ejecución y precisión, ya que mientras más frecuencias y rotaciones se utilicen, más precisa será la clasificación a coste de mayor tiempo de ejecución, y viceversa. Como consecuencia, se eligió el set de 40 *wavelets* de Gabor (complejas) de la Figura 7.9, conocido como el set de Wiskott [34] [33], y que es una solución por la que suelen optar la mayoría de los autores. En la Figura 9.10 se observa el diagrama de bloques correspondiente a la implementación del sistema de reconocimiento basado en *wavelets* de Gabor. Cabe aclarar que el pre-procesado mixto combina la ecualización de histograma y la normalización de varianza.

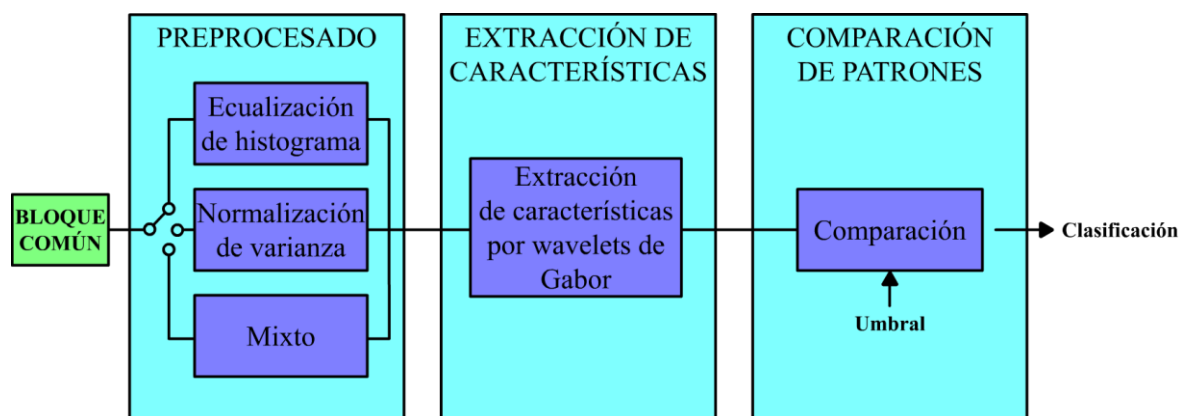


Figura 9.10. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en wavelets de Gabor. El bloque común es el mostrado en la Figura 9.1.

Propiedades

A cada una de ellas le corresponde una variable de las almacenadas en el archivo “gabor.mat” del directorio “PROPIEDADES” (Figura 9.3) y son las siguientes:

- Pre-procesado: Es el pre-procesado que será aplicado a las imágenes para el próximo entrenamiento del sistema (y consecuentemente para las posteriores clasificaciones). Éste puede ser escogido entre ecualización de histograma, normalización de varianza, o mixto.
- Puntos de la malla: Número de puntos equidistantes que tiene la malla de puntos a lo largo de la dirección del eje X y número de puntos equidistantes a lo largo del eje Y.
- Umbral de similitud.

A diferencia de los sistemas basados en PCA y LDA, el resultado de la comparación entre los patrones de entrada y los almacenados no es una distancia, sino una similitud entre vectores (ver 7.4. Reconocimiento facial basado en *wavelets* de Gabor) que varía entre 0 (nada parecidos) y 1 (idénticos). En consecuencia, se decidió que para este sistema se midiera la similitud entre un patrón de entrada y todos los patrones almacenados en la base de datos (como en el tipo de umbral por muestras), y que el usuario pueda elegir un umbral de similitud entre 0 (todos los patrones de entrada serían admitidos) y 1 (solamente sería admitido un patrón de entrada exacto al patrón almacenado).

Los valores por defecto de las propiedades para el sistema basado en *wavelets* de Gabor se pueden observar en la Tabla 9.5.

Propiedades por defecto del sistema de reconocimiento basado en <i>wavelets</i> de Gabor	
Pre-procesado	Mixto
Umbral	0,85
Puntos de la malla en el eje X	6
Puntos de la malla en el eje Y	8

Tabla 9.5. Propiedades por defecto del sistema de reconocimiento facial basado en *wavelets* de Gabor.

Datos

Para poder explicar los datos almacenados al final de cada entrenamiento del sistema que utiliza *wavelets* de Gabor es necesario entender la filosofía de trabajo que se siguió a la hora de diseñar el sistema.

El hecho de convolucionar todas y cada una de las muestras de la base de datos con las 40 *wavelets* de Gabor utilizadas para solamente almacenar los valores de dichas convoluciones en los puntos de la imagen indicados por la malla de puntos construida, supone un coste muy alto en cuanto a tiempo de entrenamiento respecta. Sin embargo, y como se comentó en el capítulo 7.4, el valor de la convolución de una imagen con una *wavelet* en un punto en específico es el mismo que el de la proyección de la misma imagen sobre la *wavelet* trasladada a dicho punto.

Como consecuencia, para la implementación del sistema de reconocimiento basado en *wavelets* de Gabor se decidió crear para cada punto de la malla las 40 *wavelets* del mismo tamaño que las imágenes de entrenamiento, pero con la onda desplazada a cada punto. Por ejemplo, dada la malla de puntos de la Figura 9.11(izquierda), en lugar de convolucionar la imagen con la *wavelet* que allí se muestra (centro) y almacenar el valor del punto superior izquierdo de la malla, se proyecta la imagen sobre la misma *wavelet* trasladada a dicho punto (Figura 9.11, derecha).



Figura 9.11. A la izquierda una malla de puntos superpuesta a una imagen, en el centro la parte real de una *wavelet* de Gabor, y a la derecha la misma *wavelet* desplazada al punto superior izquierdo de la malla. Para poder llevar a cabo la proyección, la *wavelet* trasladada debe ser del mismo tamaño que la imagen.

No está de más recordar que el gris que se observa en las imágenes en realidad corresponde con un valor cero en la *wavelet* (Figura 7.10).

Para una malla de puntos como la de la Figura 9.11, el procedimiento descrito equivale a calcular un total de 40 *wavelets* multiplicado por 24 desplazamientos de cada una (960 *wavelets*).

Dicho esto, para calcular las características de Gabor en los puntos marcados por la malla de todas las imágenes de entrenamiento, se apilan todos los vectores imagen en una matriz de observaciones X . Si de igual manera se apilan las 960 *wavelets* desplazadas como

si fueran vectores imagen en una matriz de *wavelets* GW , los coeficientes complejos de Gabor de todo el conjunto de entrenamiento (CG) se calcularían con la siguiente ecuación:

$$CG = GW^T X \quad (9.2)$$

Si se supone que las dimensiones de las imágenes de entrenamiento son $W \times H = n$, que el número de imágenes de entrenamiento es M , y que el número de *wavelets* es K , entonces las dimensiones de las matrices serían las siguientes:

$$[CG]_{K \times M} = [GW^T]_{K \times n} [X]_{n \times M} \quad (9.3)$$

Como se puede observar, es una multiplicación de matrices, por lo que el tiempo de entrenamiento se reduciría considerablemente. Por otro lado, guardar 960 *wavelets* desplazadas para entrenar el sistema supone un coste de tiempo a la hora de recuperarlas.

En consecuencia el tiempo de entrenamiento una vez generadas las *wavelets* se reduce de la convolución de cada imagen de entrenamiento con las 40 *wavelets* (muy alto y variable en función del número de imágenes de entrenamiento) al tiempo que tarde el sistema en recuperar las 960 *wavelets* y multiplicar la matriz de observaciones con la matriz de *wavelets* (moderado y prácticamente constante para cualquier número de imágenes de entrenamiento comparables).

Dicho lo anterior, los datos almacenados al finalizar cada entrenamiento del sistema basado en *wavelets* de Gabor se divide en dos conjuntos: El primer, almacenada en el archivo “datosGW.mat”; y el segundo, almacenada en el directorio “GWActuales”; ambos almacenados en el directorio “DATOS” (Figura 9.3).

Dentro del primer conjunto de datos se encuentran los siguientes:

- Los coeficientes complejos de Gabor de cada muestra resultantes de sus respectivas proyecciones sobre cada wavelet (40 coeficientes complejos por cada punto de la malla).
- Los patrones de cada muestra (los valores absolutos de cada coeficiente complejo para cada muestra).
- Tipo de pre-procesado utilizado en el último entrenamiento.

El segundo conjunto de datos resultantes está formado por un fichero llamado “condiciones.mat” y una cantidad variable de ficheros que contienen cada una de las *wavelets* desplazadas calculadas.

El fichero “condiciones.mat” contiene las dimensiones de las *wavelets* desplazadas (el mismo ancho y alto de las imágenes de la base de datos) y los puntos de la malla a lo largo del eje X y el eje Y. Gracias a este fichero, cuando se reentrena el sistema debido a un cambio en la base de datos, si se sigue manteniendo la misma malla de puntos y dimensiones de las imágenes, no es necesario volver a calcular todos los desplazamientos de las 40 *wavelets* sino que basta con recuperarlas de los otros ficheros ya comentados.

Los ficheros que contienen las *wavelets* de Gabor desplazadas son en realidad particiones de la matriz de *wavelets* GW , de tal manera que existirá uno de ellos por cada 100 *wavelets* de Gabor que contenga la matriz. Para el ejemplo de la Figura 9.11 (960 *wavelets* desplazadas) el número de ficheros será 10, donde 9 de ellos contendrán 100 *wavelets* desplazadas y el último contendrá 60 *wavelets* desplazadas.

Observando la tabla de propiedades por defecto del sistema de reconocimiento basado en *wavelets* de Gabor (Tabla 9.5), es de suponer que habrá permanentemente 20 ficheros reservados para las 1920 *wavelets* desplazadas. Dichos ficheros se encuentran dentro del directorio “GWDefecto” del directorio “PROPIEDADES_DEFECTO” (Figura 9.3).

9.9. Sistema de reconocimiento facial basado en EGM

Primeramente, se considera apropiado comentar que aunque la implementación de este sistema está enfocada al reconocimiento facial, no deja de ser una implementación de emparejamiento de grafos elásticos. Por lo tanto, si en lugar de crear una base de datos de rostros, el usuario decidiera crear una base de datos de otros objetos, la implementación seguiría siendo completamente válida.

El diagrama de bloques para este último sistema se muestra en la Figura 9.12

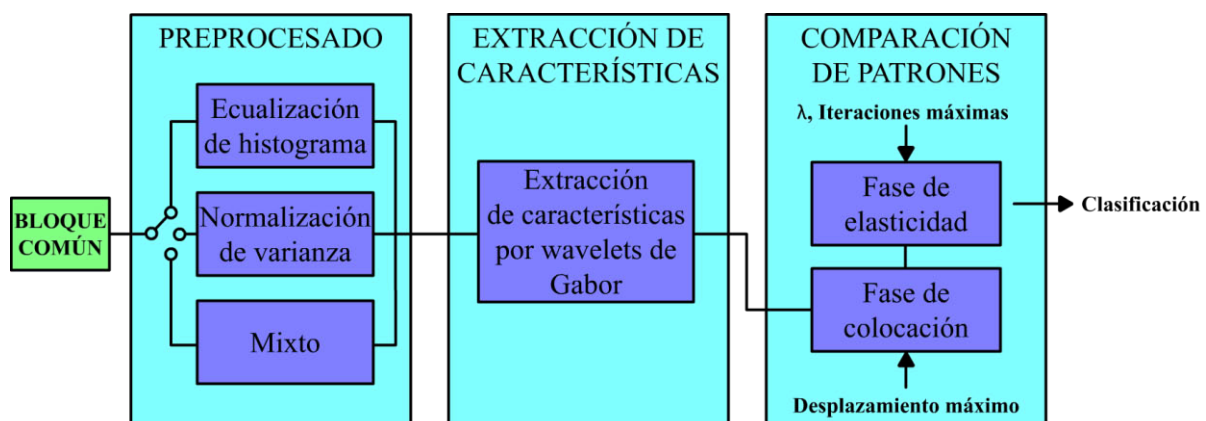


Figura 9.12. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en EGM. El bloque común es el mostrado en la Figura 9.1.

Propiedades

A cada una de ellas le corresponde una variable de las almacenadas en el archivo “EGM.mat” del directorio “PROPIEDADES” (Figura 9.3) y son las siguientes:

- El **pre-procesado** a aplicar para cada modelo. El tipo de pre-procesado especificado solamente tendrá efecto para los modelos creados después de generar un grafo con dicho pre-procesado especificado. Para poder cambiarlo es necesario volver a crear el grafo. Puede ser escogido entre ecualización de histograma, normalización de varianza, o mixto.
- El **desplazamiento** del barrido del grafo en la etapa de colocación.

- El **factor de rigidez** (λ) y el **número de iteraciones máximas** sin mejorar el coste de la etapa de elasticidad. El factor de rigidez puede ser infinito.

Las propiedades por defecto pueden ser observadas en la Tabla 9.6.

Propiedades por defecto del sistema de reconocimiento basado EGM	
Pre-procesado	Mixto
Desplazamiento de la etapa de colocación	10
Factor de rigidez	0,005
Iteraciones máximas sin reducir el coste de la etapa de elasticidad	20

Tabla 9.6. Propiedades por defecto del sistema de reconocimiento facial basado en EGM.

Es importante destacar que el directorio “PROPIEDADES” contiene también un directorio llamado “GWEGM” que contiene un archivo con las 40 *wavelets* de Gabor utilizadas (Figura 7.9), con un tamaño de 128x128 píxeles cada una. Esto se debe a que la ubicación de los vértices en este sistema varía en función del modelo, lo que conduce a que sea inútil el uso de la proyección en lugar de la convolución como se hizo en el sistema basado en *wavelets* de Gabor.

Datos

Para el caso del reconocimiento facial por emparejamiento de grafos elásticos, los datos se almacenan de una forma distinta, ya que se toma como entrenamiento el hecho de generar un modelo para un individuo. Es evidente que un individuo puede tener varios modelos en una implementación de EGM, sobre todo si se consideran poses con deformaciones extremas [31]. No obstante, y por cuestiones de simplicidad, para esta implementación se optó porque cada individuo tenga un único modelo.

Con el fin de que el usuario pueda crear grafos personalizados, se programó en MATLAB una herramienta interactiva que permite añadir vértices y conectarlos con aristas a partir de una imagen de ejemplo. En la Figura 9.13 se observa la interfaz gráfica de dicha herramienta.



Figura 9.13. Herramienta de creación de grafos.

La misma herramienta de creación de grafos propone además la creación de grafos predefinidos que consisten en una rejilla de puntos similar a la malla de puntos de las *wavelets* de Gabor, donde cada punto es un vértice que está conectado con sus vecinos verticales y horizontales (Figura 9.14, izquierda). Sin embargo, se permite también editar el grafo predefinido para crear otras formas a partir de la rejilla, como se puede observar en la Figura 9.14 a la derecha.

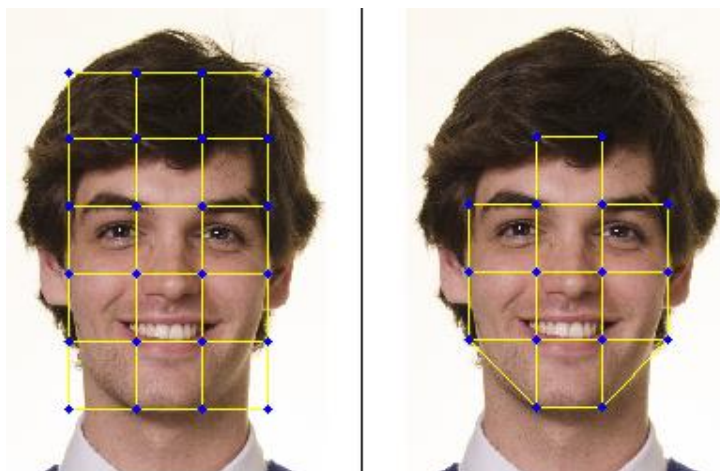


Figura 9.14. Ejemplos de un grafos predefinidos creados con la herramienta de creación de grafos programada.

Una vez generado el grafo, la creación de modelos para cada individuo se lleva a cabo con una herramienta (también programada durante la elaboración de esta implementación) que indica al usuario en qué puntos deben estar los vértices para crear un modelo con la misma forma del gafo creado. Dicha herramienta se puede observar en la Figura 9.15.



Figura 9.15. Herramienta de creación de modelos.

Cuando la creación de un grafo se inició de forma predefinida, la herramienta de generación de modelos genera el modelo de forma automática para evitar problemas de precisión en la rectitud de las aristas.

Una vez finalizada la elaboración de un grafo, se guardan en el fichero “datosEGM.mat” del directorio “DATOS” (Figura 9.3) las siguientes variables:

- Pre-procesado especificado antes de la creación del grafo. Éste será el pre-procesado aplicado a una imagen antes de generarle un modelo (y en consecuencia el mismo que será aplicado en clasificación).
- Una variable de tipo “flag” que indica si el grafo fue creado de forma predefinida.
- Lista de la ubicación de los vértices en la imagen de ejemplo para que la herramienta de creación de modelos pueda recrear la creación del grafo en la imagen de ejemplo. En el caso de grafos predefinidos, las posiciones de los vértices se guardan normalizadas con respecto a las dimensiones de la imagen para permitir la creación automática de modelos.
- Lista de aristas que conectan a cada vértice.

Las variables que contienen la ubicación de los vértices y de las aristas que conectan a cada vértice son matrices como las siguientes:

$$Lista\ de\ vértices = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

$$Lista\ de\ aristas = \begin{bmatrix} v_{1i} & v_{1f} \\ v_{2i} & v_{2f} \\ \vdots & \vdots \\ v_{mi} & v_{mf} \end{bmatrix}$$

Donde n y m son el número de vértices y aristas respectivamente, (x_j, y_j) son las coordenadas del vértice v_j , y (v_{ki}, v_{kf}) son los vértices inicial y final que conecta la arista k .

Luego de haber generado un grafo, el siguiente paso es crear un modelo para cada individuo. Cada modelo es almacenado en el directorio “INDIVIDUOS-EGM” (Figura 9.3) como un archivo con nombre “ID[X]-[APELLIDOS] [NOMBRE].mat”. Cada archivo modelo contiene la siguiente información:

- Lista de ubicación de los vértices en la imagen desde la cual se generó el modelo. Su forma es igual a la de la lista de vértices para el grafo de ejemplo.
- Lista de etiquetas de las aristas. Consiste en una matriz como la siguiente:

$$Lista\ de\ etiqueta\ de\ aristas = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_m & y_m \end{bmatrix}$$

Es importante recordar que la etiqueta de cada arista es un vector distancia. En consecuencia, cada (x_k, y_k) de la matriz son las coordenadas del vector distancia \vec{d}_k .

- Lista de jets (etiquetas de las aristas). Es de la forma:

$$Lista\ de\ jets = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,40} \\ C_{2,1} & C_{2,2} & \dots & C_{2,40} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n,1} & C_{n,2} & \dots & C_{n,40} \end{bmatrix}$$

Donde cada J_j es el jet correspondiente al vértice v_j y cada $C_{j,g}$ es una característica de Gabor de las incluidas en dicho jet.

Adicionalmente a los modelos de cada individuo, en el directorio “INDIVIDUOS-EGM” de la Figura 9.3 se almacenan también dos ficheros llamados “etiquetasEGM.mat” (matriz de etiquetas de EGM) y “actualización.mat” (matriz de actualización de EGM).

En la matriz de etiquetas de EGM se enlaza cada modelo a la identidad de un individuo de forma similar que en la matriz de etiquetas de la base de datos (ver capítulo 9.3), solamente que en lugar de ser la tercera columna de la matriz el número de imágenes de cada individuo, ésta representa la imagen del individuo para la cual se generó el grafo.

En la matriz de actualización de EGM se guarda un historial con las acciones de edición y eliminación de individuos de la base de datos idéntico al del historial de clasificación. Una función es encargada de verificar si alguna operación puede afectar al sistema de clasificación por grafos elásticos. Por ejemplo, si se cambia el nombre de un individuo, se cambia también el nombre de su modelo; o si se elimina la imagen de un individuo con la que se creó su modelo, se elimina su modelo.

En vista de la demora del método de EGM para llevar a cabo una clasificación y que los costes resultantes son muy variables dependiendo de la base de datos utilizada, condiciones de captura, grafos generados, etc. se optó por no trazar ningún umbral de decisión, convirtiendo al sistema de clasificación basado en EGM en un sistema que trabaja única y exclusivamente en modo identificación cerrado.

9.10. Ventanas de visualización

Con el objetivo de dar al usuario una realimentación visual de los resultados del entrenamiento, cada sistema está dotado con una ventana de visualización. Dependiendo del sistema, en dicha ventana se puede observar:

- PCA: La cara media y las *eigenfaces* calculadas junto con el porcentaje acumulado información que se representa hasta cada una de ellas.
- LDA: La cara media y las *fisherfaces* calculadas.
- *Wavelets* de Gabor: Las *wavelets* de Gabor utilizadas y la malla de puntos construida. Debe recordarse que para esta implementación las *wavelets* de Gabor siempre serán las mismas.
- EGM: Las *wavelets* de Gabor utilizadas.

Por ejemplo, en la Figura 9.16 se puede observar la ventana de visualización para el sistema de reconocimiento facial basado en PCA.

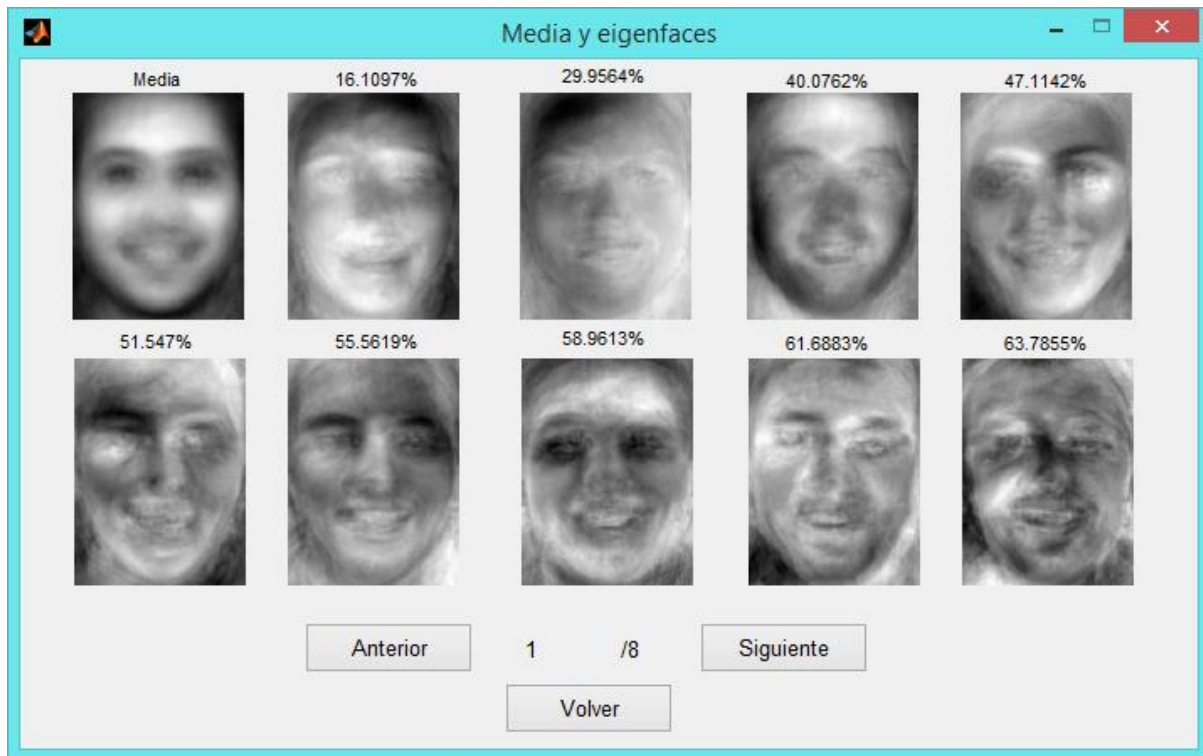


Figura 9.16. Ventana de visualización para el sistema de reconocimiento facial basado en PCA. Las eigenfaces que en ella se muestran son las correspondientes a una base de datos personal del autor.

9.11. Reconocimiento facial en modo identificación

Se puede acceder a la clasificación en modo identificación desde cualquiera de los 4 sistemas (Figura 9.2), dando a entender al programa que la técnica en la que se basa dicho sistema es la que se va a utilizar para la clasificación. Cabe destacar que independientemente de la técnica a utilizar, el programa se limitará a dar una lista de coincidencias de únicamente 4 individuos.

Para llevar a cabo una clasificación en modo identificación, el usuario debe proporcionar al programa una imagen de entrada (desde archivos, o desde cámara web). Posteriormente, y como se puede observar en la Figura 9.17, se muestra al usuario el individuo de la base de datos más parecido al de la imagen de entrada (el primero de la lista de coincidencias).

Si el usuario está conforme con la clasificación, puede indicarle al sistema para incluir dicha imagen de entrada en el historial. De forma contraria, si el usuario no está conforme con la clasificación, debe indicárselo al sistema y éste le mostrará las siguientes 3 mejores coincidencias. Si el individuo en cuestión se encuentra dentro de esa segunda lista, el usuario tiene la posibilidad de seleccionar la identidad correcta para añadirla al historial. En el caso de que el individuo en cuestión no se encuentre en la segunda lista, se considera que el sistema ha fallado completamente, y el programa le mostrará al usuario una lista con todas las posibilidades para que el usuario le indique la identidad correcta y ésta sea añadida al historial.



Figura 9.17. Ventana de clasificación mostrando la primera coincidencia de la lista de coincidencias.

El programa clasificará a un individuo como desconocido siempre que la lista de coincidencias tenga un número de coincidencias menor de cuatro individuos. Por ejemplo, si en la lista de coincidencias se encuentra únicamente un individuo de la base de datos, se considera la segunda opción como un desconocido; o si la lista de coincidencias tiene a dos individuos de la base de datos, la tercera opción será un desconocido (Figura 9.18, arriba).

En el caso de que la lista de coincidencias esté vacía, la única opción que se presenta al usuario es la de individuo desconocido (Figura 9.18, abajo).

Se considera importante comentar un problema que surgió durante la clasificación de un individuo utilizando la técnica de emparejamiento por grafos elásticos. Dicho problema radicaba en el hecho de que el proceso de clasificación de EGM tiene la desventaja de ser un proceso de colocación de vértices aleatorio (ver capítulo 8.2), haciendo difícil la posibilidad de mostrar por pantalla una realimentación que permita al usuario conocer el estado de la clasificación.

El uso de una barra de espera no es viable en este caso, puesto que el tiempo de procesado es aleatorio. Por otro lado, el uso un mensaje de espera resulta incierto, ya que al ser un tiempo de espera indeterminado puede prestarse a confusión con un bloqueo del proceso en el caso de que tome demasiado tiempo la clasificación.

Dicho esto, se propuso una solución original que solventa el problema de una manera elegante: Mostrar al usuario de manera gráfica las fases de colocación y elasticidad de emparejamiento. De esta forma el usuario no solamente tendrá la certeza de que el programa no está bloqueado, sino que además conocerá el estado exacto de la clasificación.



Figura 9.18. Ejemplos donde el programa clasifica al individuo de la imagen de entrada como un desconocido.

Por último, no está de más comentar que para la implementación del sistema basado en EGM, en la ventana de clasificación del modo identificación permite observar el grafo deformado (Figura 9.19).

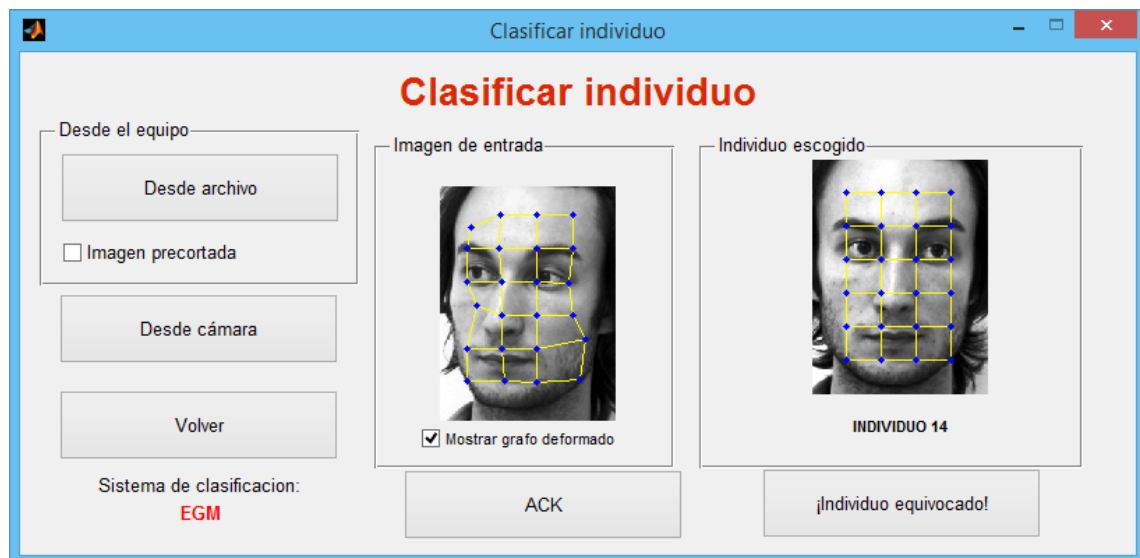


Figura 9.19. Ventana de clasificación en modo identificación para el sistema basado en EGM.

9.12. Reconocimiento facial en modo verificación

Al igual que en el modo identificación, se puede acceder a la clasificación en modo verificación desde cualquier sistema a excepción del sistema basado en EGM, que como ya se comentó anteriormente, no se implementó en modo verificación (Figura 9.2).

Para este modo de clasificación, en lugar de mostrar los primeros individuos de una lista de coincidencias, se pide al usuario que se identifique y que luego proporcione una imagen de entrada. Una vez que el programa calcula el patrón de entrada, éste es comparado con el patrón almacenado en la base de datos de la identidad proporcionada al principio de la operación (o los patrones, para el caso del umbral por muestras). En el caso de que el patrón de entrada se encuentre dentro de los límites establecidos por el umbral, el individuo será clasificado como legítimo, o como impostor en el caso contrario. En la Figura 9.20 se puede observar el resultado de una clasificación en modo verificación del programa implementado.



Figura 9.20. Ventana de clasificación del modo verificación.

Aunque la imagen de entrada puede ser proporcionada desde archivo o por cámara web, es evidente que para desarrollar una aplicación real la opción de proporcionar una imagen desde archivo debería eliminarse (los sistemas en modo verificación se suelen utilizar para controlar accesos). No obstante, el programa resultante de este proyecto la mantiene para los casos en los que se pretenda usar con fines educativos o de investigación.

9.13. Propuesta de un modo híbrido de clasificación

En vista de los buenos resultados obtenidos y del poco tiempo que consume una clasificación en modo identificación para los sistemas basados en PCA, LDA y *wavelets* de Gabor, se propone un modo híbrido de verificación con el objetivo de reducir las tasas de falsa aceptación del modo de verificación tradicional y aumentar el rechazo frente a individuos externos a la base de datos (**aumentar la seguridad** de un sistema de reconocimiento):

Un individuo es únicamente clasificado como legítimo si se encuentra en la primera posición de la lista de coincidencias del modo identificación.

Se recomienda que el modo híbrido sea utilizado con umbrales estrechos para aumentar a máximo el rechazo del sistema frente a individuos externos a la base de datos, aunque no tan estrechos como para alcanzar altas tasas de falso rechazo.

Para terminar, es evidente que podrían definirse otros modos híbridos que utilicen las demás posiciones de la lista de coincidencias. No obstante, estos modos se desviarían del objetivo principal (reducir las tasas de falsa aceptación del modo de verificación tradicional).

9.14. Implementación de un programa de evaluación

Adicionalmente al programa que abarca los 4 sistemas de reconocimiento facial estudiados, la implementación cuenta con un programa de evaluación que permite graficar las curvas de evaluación (ver capítulo 2.3) de cada uno de los sistemas, y con las propiedades que se deseen. En consecuencia, la estructura de diseño del programa original de la Figura 9.3 quedaría como la de la Figura 9.21.

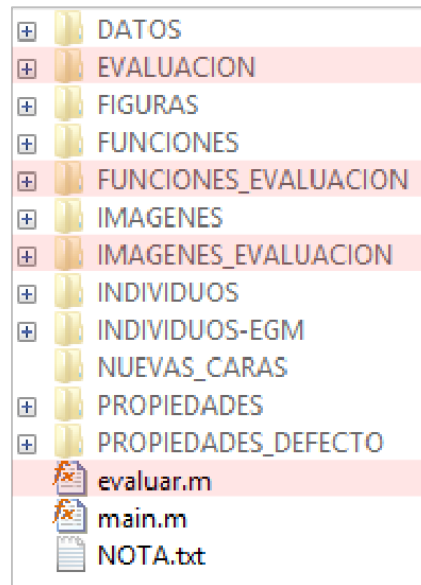


Figura 9.21. Estructura de diseño del programa con los directorios y función principal del programa de evaluación incluidos.

La nueva estructura de diseño cuenta con la función “evaluar”, función principal del programa de evaluación; el directorio “FUNCIONES_EVALUACION”, que contiene todas las funciones y figuras necesarias para evaluar cada sistema; el directorio “EVALUACION”, que contiene los archivos resultantes de cada evaluación realizada; y el directorio “IMAGENES_EVALUACION”, que contiene las imágenes que forman parte del conjunto de validación con el que se desea evaluar los sistemas.

El menú principal del programa de evaluación se muestra en la Figura 9.22. En él se puede observar que la lista de evaluaciones realizadas utiliza una nomenclatura especial. Dicha nomenclatura, independientemente del sistema evaluado, sigue el siguiente formato:

[Sistema]-[Propiedad 1]-[Propiedad 2]-[...] -[Propiedad N]

En la Tabla 9.7 se muestran las propiedades especificadas dentro de la nomenclatura para cada sistema.

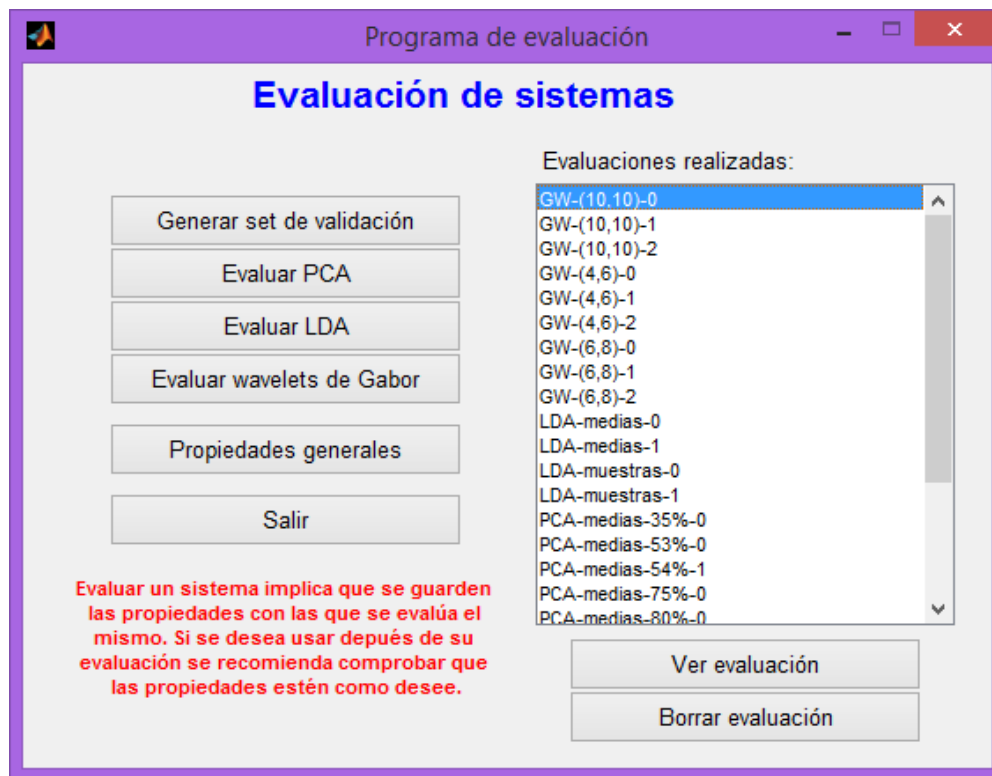


Figura 9.22. Menú principal del programa de evaluación.

Técnica	Sistema	Propiedad 1	Propiedad 2	Propiedad 3
Análisis por componentes principales	PCA	medias - Para umbral por medias	Resolución	0 - Para ecualización de histograma
		muestras - Para umbral por muestras		1 - Sin pre-procesado
Análisis lineal discriminante	LDA	medias - Para umbral por medias	0 - Para ecualización de histograma	-
		muestras - Para umbral por muestras	1 - Sin pre-procesado	
Wavelets de Gabor	GW	(PX,PY)	0 - Para ecualización de histograma	-
			1 - Para normalización de varianza	
			2 - Mixto	

Tabla 9.7. Nomenclatura del programa de evaluación.

El procedimiento a seguir para llevar a cabo la evaluación de un sistema de reconocimiento facial con el programa de evaluación es el siguiente:

1. Generar un conjunto de evaluación formado por una imagen facial de cada individuo de la base de datos (Se facilita una herramienta para hacerlo al pulsar el botón “Generar set de evaluación” que se observa en la Figura 9.22).
2. Evaluar cualquiera de los tres sistemas (el programa de evaluación permite al usuario elegir las propiedades que mejor considere antes de evaluar un sistema).
3. Seleccionar de la lista la evaluación que se desee visualizar.

Es importante comentar un problema que surge cuando se desean graficar las curvas ROC y DET cuando un sistema de reconocimiento es evaluado. Dicho problema reside en el hecho de que el número de individuos legítimos para cualquier sistema de clasificación es muchísimo menor que el número de individuos impostores.

Una solución por la que se suele optar es interpolar los datos de los individuos legítimos. Sin embargo, en lugar de utilizar interpolación, se optó por aproximar las densidades de probabilidad de individuos legítimos e impostores a alguna densidad de probabilidad conocida. Consecuentemente, cada vez que el usuario desee visualizar una evaluación, se mostrará una ventana como la de la Figura 9.23 donde se permite al usuario elegir la estimación que mejor considere.

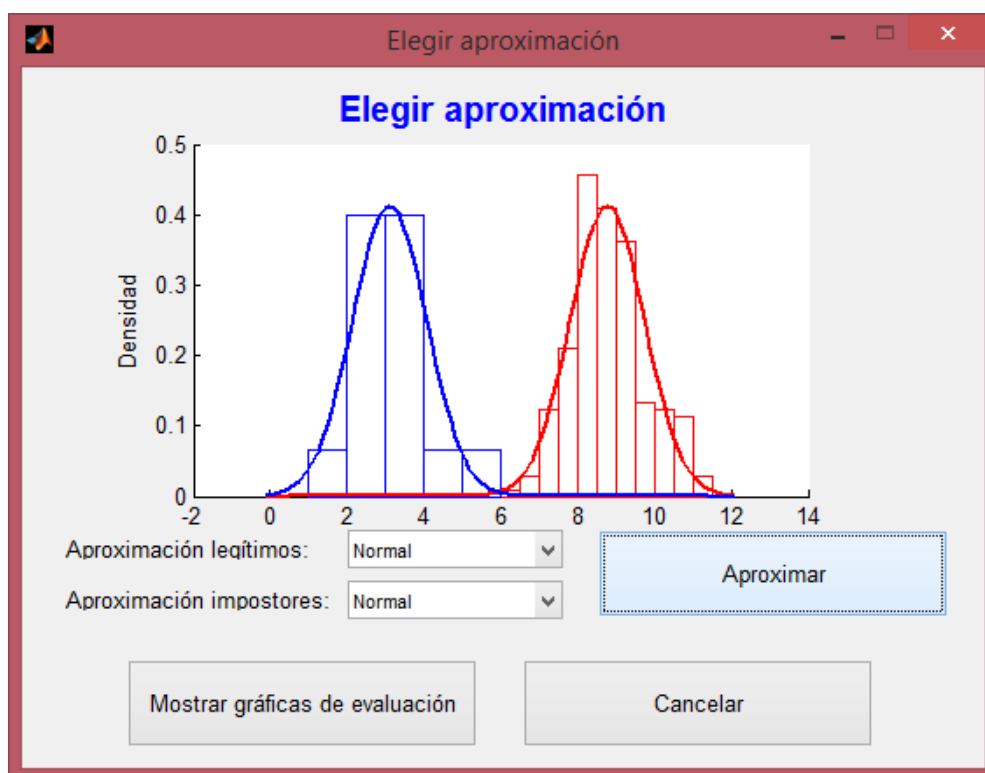


Figura 9.23. Ventana de elección de la estimación de densidad de probabilidad.

Una vez elegida la estimación deseada, el programa mostrará las siguientes curvas: densidades de probabilidad de individuos legítimos e impostores, curvas FAR y FRR en función del umbral de decisión, ROC, y DET.

Como ya se comentó, las curvas ROC y DET son calculadas a partir de la estimación seleccionada. No obstante, las curvas de FAR y FRR en función del umbral de decisión sí pueden ser calculadas a partir de los datos empíricos. Dicho esto, se decidió que el programa de evaluación muestre ambas: las curvas obtenidas a partir de la estimación escogida, y las curvas obtenidas a partir de los datos empíricos. En la Figura 9.24 se pueden observar unas curvas de FAR y FRR en función del umbral de decisión. Las curvas obtenidas de forma empírica son representadas como líneas finas, mientras que las obtenidas mediante estimaciones son representadas como líneas gruesas.

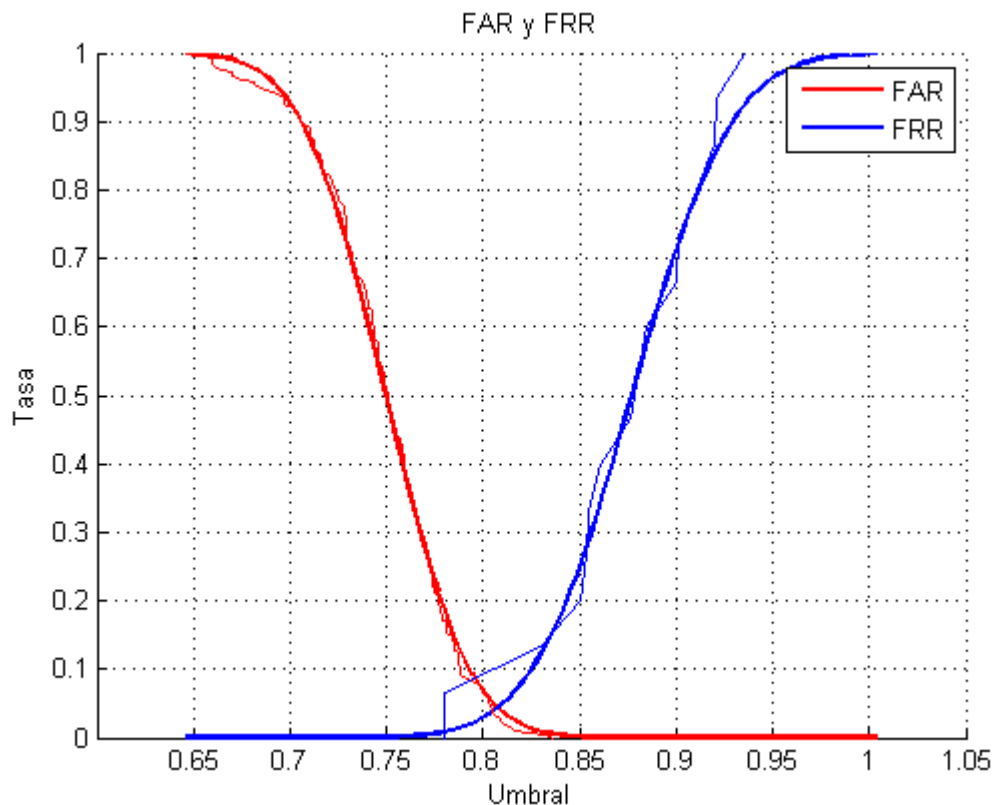


Figura 9.24. Curvas FAR y FRR en función del umbral resultantes de un sistema. El programa de evaluación representa las curvas obtenidas de forma empírica como líneas finas, y las obtenidas mediante estimaciones como líneas gruesas.

Otra característica significativa con la que se dotó el programa de evaluación es que si se seleccionan varias evaluaciones (hasta un máximo de 4) se mostrarán en una sola gráfica todas las ROC de las evaluaciones seleccionadas y en otra gráfica todas las DET. Un ejemplo de una gráfica ROC de este tipo puede observarse en la Figura 9.25.

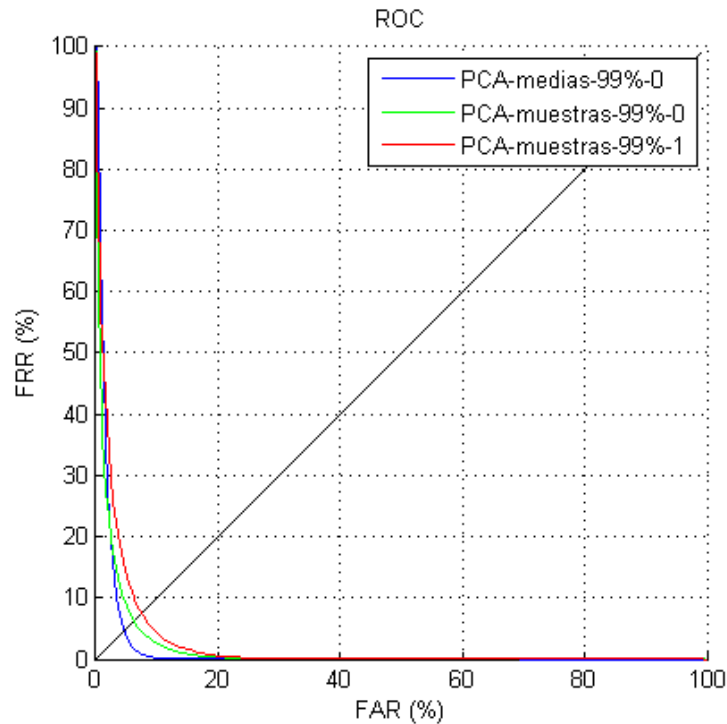


Figura 9.25. Múltiples curvas ROC mostradas en una sola gráfica por el programa de evaluación.

Finalmente, se recuerda que las curvas de evaluación de un sistema de reconocimiento son calculadas a partir de su rendimiento en el modo verificación, siendo este el motivo por el cual no aparece el sistema de reconocimiento facial basado en EGM dentro del programa de evaluación.

Para más información de cómo utilizar el programa de evaluación, en el manual de uso incluido en los anexos se muestra también una guía de cómo utilizar el programa de evaluación.

Antes de pasar al capítulo de presentación de los resultados, es conveniente comentar que el código fuente en MATLAB de tanto el programa con los 4 sistemas de reconocimiento facial, como su respectivo programa de evaluación, están disponibles en el DVD anexo a este documento.

PARTE IV: RESULTADOS, CONCLUSIONES Y LÍNEAS FUTURAS

10.Resultados

Como se ha podido comprobar en el capítulo 9, cada uno de los sistemas de reconocimiento propuestos en el programa posee una serie de propiedades modificables a gusto del usuario. En consecuencia, no sería productiva una presentación de los resultados que combine todas las posibilidades que ofrece cada sistema. No obstante, se eligió un conjunto de configuraciones de cada sistema que permite contrastarlas entre ellas y dar una idea de las posibilidades que se ofrecen.

Es importante destacar que los resultados que se presentan en este capítulo para cada base de datos son única y exclusivamente válidos para los conjuntos de entrenamiento formados, con los conjuntos de validación utilizados, y bajo las condiciones especificadas. El problema del reconocimiento facial es un tema muy general que depende de muchos parámetros, y por lo tanto, los resultados no deben ser generalizados.

Para llevar a cabo la evaluación de cada sistema, se utilizaron 3 bases de datos:

- Base de datos de *Yale University* [21]: Está compuesta por 11 imágenes frontales de 15 individuos con diferentes expresiones faciales y cambios en la dirección de la luz de algunas imágenes (Figura 5.4). Se puede decir que las condiciones de adquisición son medianamente controladas para esta base de datos. Como conjunto de entrenamiento se utilizaron 10 de las 11 imágenes de cada individuo, y como conjunto de evaluación se utilizó para cada individuo la imagen sobrante.
- Base de datos del *Centro Universitário da FEI* [37]: Está compuesta por 15 imágenes de 200 individuos con diferentes ángulos de rotación horizontal de la cabeza, una imagen sonriendo, y una con condiciones de iluminación cambiadas (generalmente baja iluminación). Para el conjunto de entrenamiento se utilizaron 5 de las imágenes de solamente 15 individuos y para el de evaluación se utilizó la imagen con condiciones de iluminación cambiadas para cada uno de ellos. Se puede decir que las condiciones de adquisición son medianamente controladas.
- Base de datos personal no-controlada: Se trata de una base de datos formada por 6 imágenes de 15 individuos (compañeros del autor de este proyecto) extraídas de redes sociales. Como conjunto de entrenamiento se escogieron las 5 imágenes que se consideraba que favorecerían más el reconocimiento (Por el ángulo de inclinación de la cabeza, condición de iluminación, nitidez de la imagen, etc.) y como conjunto de validación se escogió la imagen más desfavorable. Las condiciones de adquisición para esta base de datos no son controladas en lo absoluto.

Para cada base de datos se compararon diferentes configuraciones (combinaciones de propiedades) de cada uno de los tres sistemas implementados en modo verificación mediante sus curvas ROC y DET. De los resultados obtenidos para cada sistema, se escogió una configuración **representante** correspondiente con la configuración que mejores resultados obtuvo. Dicha configuración fue utilizada finalmente para ser comparada con las representantes de las otras técnicas y así poder extraer conclusiones.

Finalmente, y antes de comenzar con la presentación de los resultados, cabe destacar que todas las gráficas mostradas a lo largo de este capítulo fueron obtenidas con el programa de evaluación implementado utilizando la estimación por el método Kernel de las densidades de probabilidad. Esto se debe a que dicha distribución es más fiel a la verdadera distribución de cada conjunto de entrenamiento a pesar de que la distribución ideal sería una distribución normal. Otro detalle que debe tomarse en cuenta, es que la nomenclatura que aparece en la leyenda de todas las curvas ROC y DET es la misma que la explicada en la Tabla 9.7.

10.1. Curvas de evaluación sobre la base de datos de Yale University

Para obtener las imágenes de entrenamiento, los rostros fueron segmentados de manera manual seleccionando la parte de la cara limitada por las cejas (límites laterales y superior) y barbilla (límite inferior) y con los parámetros de corte adicional indicados en la Tabla 10.1.

Porcentajes de corte adicional (%)	
Porcentaje lateral	15
Porcentaje superior	30
Porcentaje inferior	5

Tabla 10.1. Porcentajes de corte adicional utilizados para la segmentación de rostros de la base de datos de Yale University

En la Figura 10.2 se puede observar un ejemplo del corte utilizado.

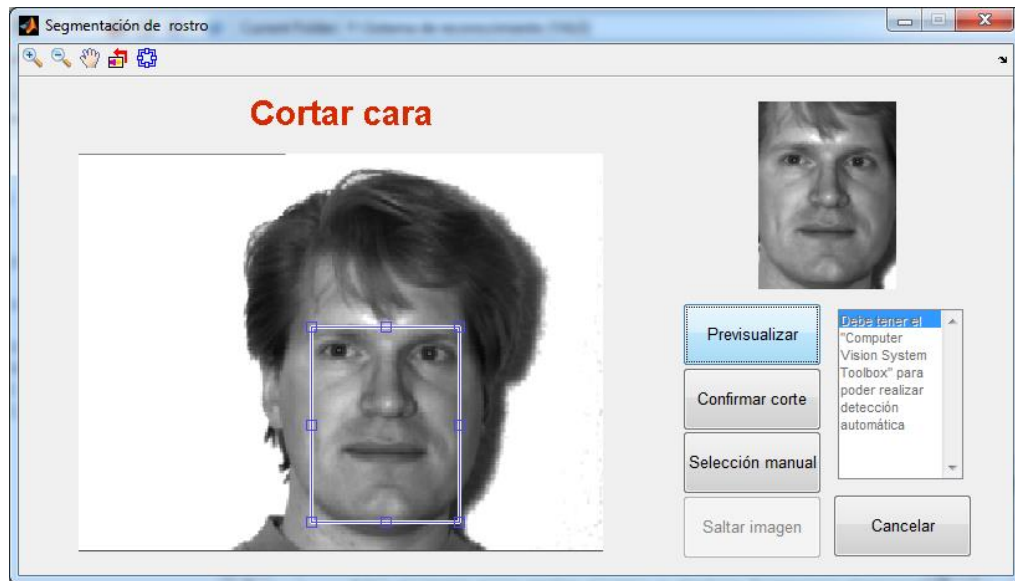


Figura 10.1. Tipo de corte utilizado para segmentar la imágenes de la base de datos de Yale University.

PCA

Entre las propiedades que se pueden cambiar en el sistema basado en PCA se encuentran el tipo de umbral, el pre-procesado, y la resolución (porcentaje de información que representan las *eigenfaces*).

La primera comparación a evaluar fue el comportamiento del sistema para todas las combinaciones posibles de tipos de umbral con tipos de pre-procesado a máxima resolución, y en función de los resultados se eligió la mejor combinación para llevar a cabo una segunda comparación a diferentes resoluciones.

En la Figura 10.2 se pueden observar las ROC y DET de la primera comparación comentada. De ella se puede deducir que **es posible** aumentar la precisión de un sistema de clasificación mediante la aplicación de un pre-procesado adecuado (ecualización de histograma en este caso).

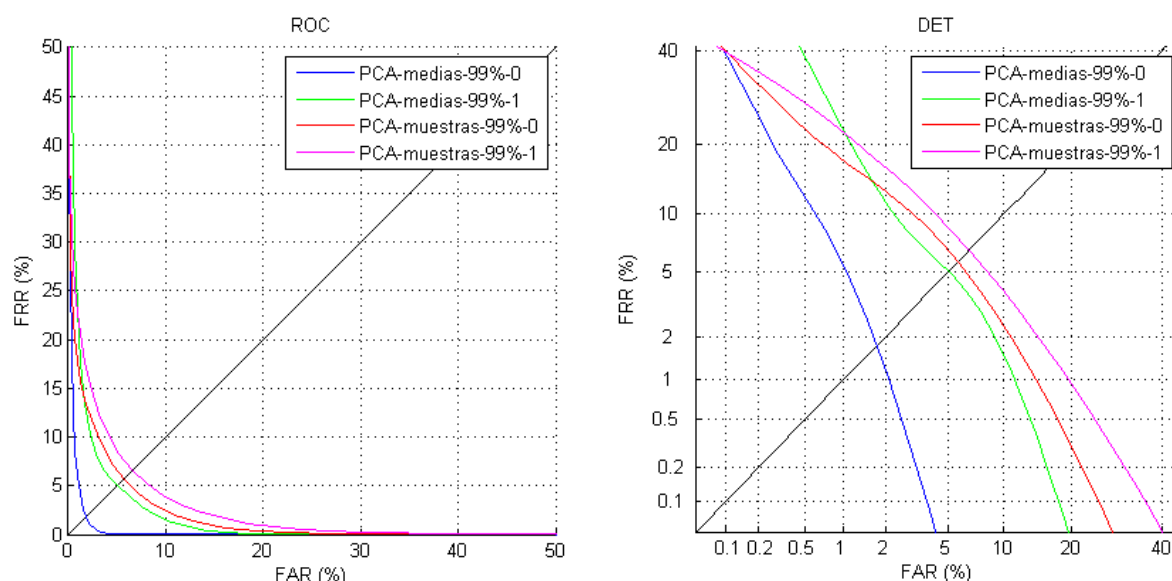


Figura 10.2. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral a una resolución del 100% (BD: YALE).

Es importante destacar la forma de las curvas DET. Nótese que para el umbral por medias sin pre-procesado resulta una curva con varios cambios de pendiente, mientras que para la curva del umbral por muestras sin pre-procesado, la forma se aproxima más a una línea recta. El hecho de que una DET se aproxime bastante a una línea significa que la distribución de los datos en la zona que se muestra en la DET es muy parecida a una distribución normal. Si en lugar de utilizar una estimación de distribución por el método Kernel, se utilizara una estimación normal de la distribución de los datos, todas las DET tendrían forma de una línea recta [5].

También es importante señalar que aunque en las leyendas se indique que se utilizó un sistema con 99% de resolución, en realidad se utilizó un sistema con 100% de resolución (todas las *eigenfaces*). El motivo por el que allí muestre 99% es que dicha resolución es la resolución real actualizada calculada por MATLAB a través de una división con una precisión muy alta redondeada al menor número entero más cercano para ser mostrada al usuario (ver capítulo 9.6).

Por otro lado, se puede decir que la mejor configuración se consiguió utilizando un umbral por medias con ecualización de histograma como pre-procesado, y por lo tanto, dicha configuración fue utilizada para llevar a cabo la segunda comparación comentada (cambio de resoluciones). Las resoluciones mínimas consideradas fueron 75%, 50% y 20%, siendo las resoluciones actualizadas 75%, 53% y 35%, respectivamente. En la Figura 10.3 se muestran las ROC y DET de esta segunda comparación, donde también se muestra como referencia la curva con resolución máxima.

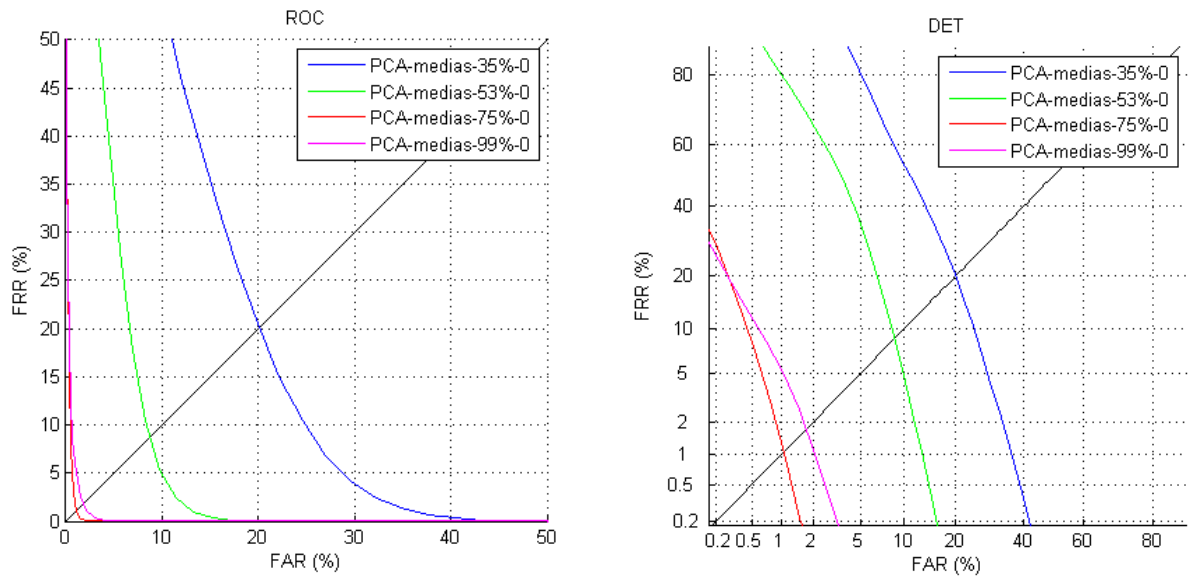


Figura 10.3. Curvas ROC y DET para distintas resoluciones de un umbral por muestras y con ecualización de histograma como pre-procesado (BD: YALE).

De manera inesperada, en las gráficas de la Figura 10.3 se observa que el sistema con resolución del 75% posee una mejor precisión que el sistema con una resolución del 100%. Esto sugiere para las condiciones especificadas, lo ideal no sería utilizar la máxima resolución disponible en PCA, sino una resolución menor a la máxima. Con el fin de hallar una resolución óptima, se repitió el procedimiento de comparación de resoluciones para las siguientes resoluciones: 75%, 80%, 85% y 100% (Figura 10.4).

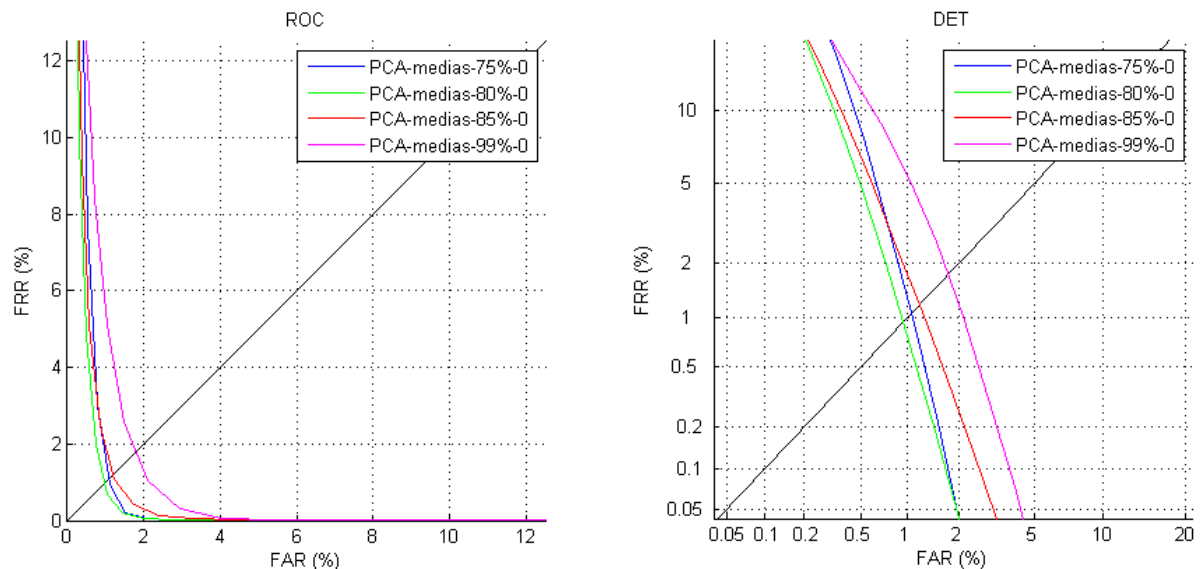


Figura 10.4. Curvas ROC y DET para las resoluciones mínimas: 75%, 80%, 85% y 100% (BD: YALE).

De la Figura 10.4 se deduce que la resolución óptima del sistema de reconocimiento basado en PCA bajo las condiciones especificadas se encuentra alrededor del 80% (se recuerda que la resolución real es ligeramente superior).

Cabe también destacar que para dicha resolución mínima de 80%, el número de *eigenfaces* se suele reducir significativamente. Para este caso en específico, una resolución mínima del 80% corresponde a utilizar tan solo las 21 primeras *eigenfaces* de las 150 calculadas.

Para terminar con la evaluación del sistema en PCA, se decidió hacer una última comparación para conocer el rendimiento de las distintas configuraciones del sistema fijando una baja resolución. En la Figura 10.5 se observan las curvas correspondiente a las distintas configuraciones posibles para una resolución mínima del 50%. Nótese que los porcentajes de varianza total representados por las *eigenfaces* dependen del pre-procesado aplicado a las imágenes (ver leyendas de la figura).

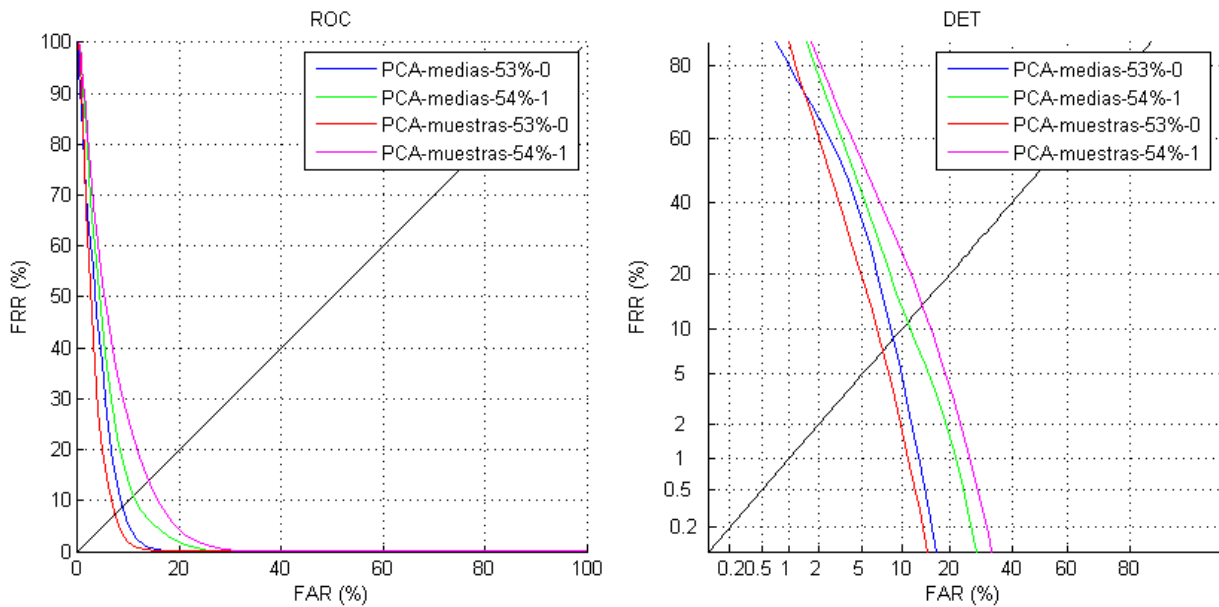


Figura 10.5. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral a una resolución mínima del 50% (BD: YALE).

En la Figura 10.5 se observa que el umbral por muestras es más robusto a resoluciones pequeñas que el umbral por medias siempre y cuando se le haya aplicado un pre-procesado que lo favorezca.

Presentados finalmente todos los resultados del sistema basado en PCA para la base de datos de *Yale University*, se toma como representante para comparar con los otros sistemas la configuración corresponde a un tipo de umbral por medias, con ecualización de histograma, y a un 80% de resolución. Las curvas de densidad de probabilidad de legítimos e impostores, y de FAR y FRR en función del umbral de dicha configuración se muestran en la Figura 10.6.

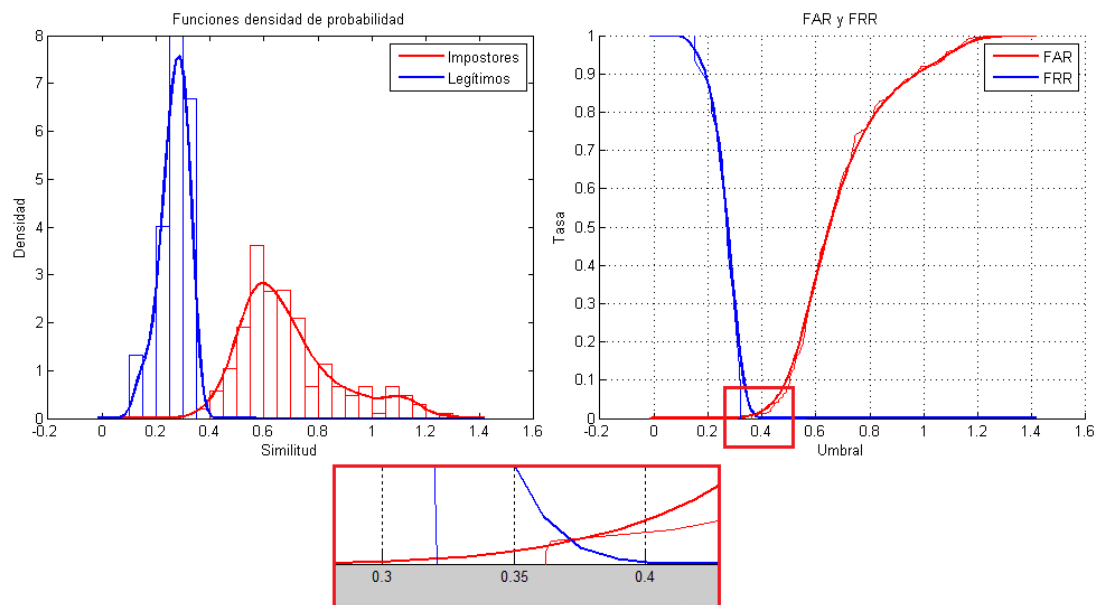


Figura 10.6. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en PCA (BD: YALE).

Nótese que a pesar de que las curvas de FAR y FRR calculadas a partir de las estimaciones se cruzan, los datos empíricos quedan perfectamente separados, consiguiendo que para las condiciones específicas y para las imágenes de validación utilizadas se pueda trazar un umbral que consiga un 0% de falso rechazo y un 0% de falsa aceptación. Esto en ningún caso significa que el sistema sea perfecto en todos los casos, sin embargo, podría afirmarse que no es un mal sistema de clasificación.

LDA

En vista de que para el sistema basado en LDA no se define ninguna resolución, simplemente se compararon todas las configuraciones posibles (Figura 10.7). Nótese que para LDA, utilizar la ecualización de histograma como pre-procesado solamente implica una mejora para el caso de umbral por muestras, mientras que para el umbral por medias no es así. Nótese también que a pesar de dicha mejora se sigue prefiriendo un umbral por medias.

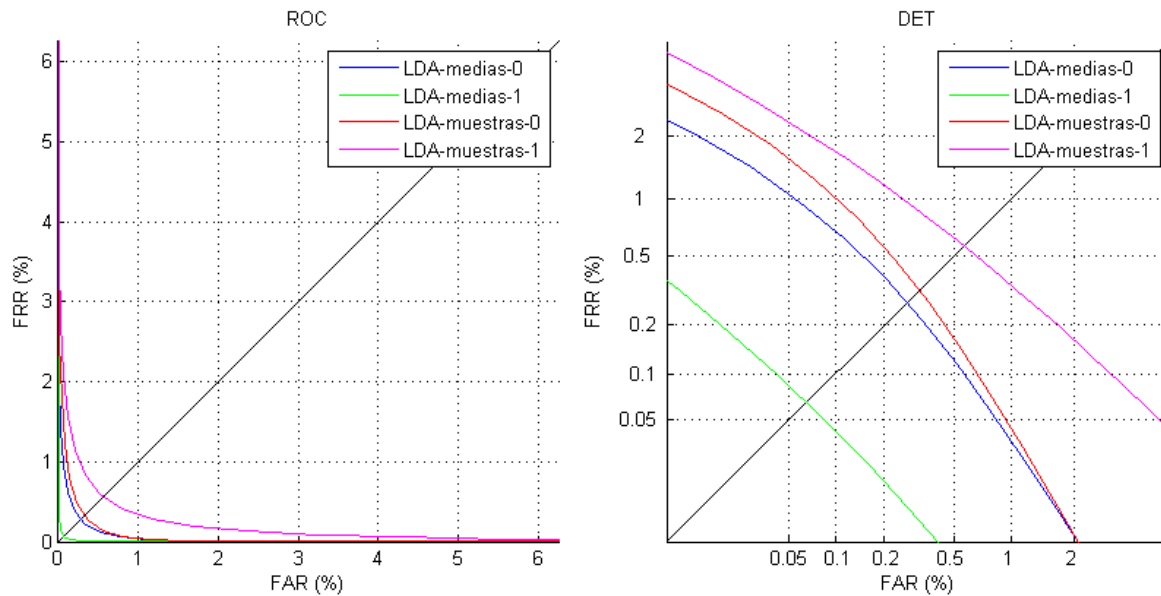


Figura 10.7. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral para un sistema basado en LDA (DB:YALE).

La mejor configuración en este caso resultó ser un tipo de umbral por muestras sin pre-procesado. En la Figura 10.8 se muestran las densidades de probabilidad de individuos legítimos e impostores y las curvas FAR y FRR en función del umbral para esta configuración.

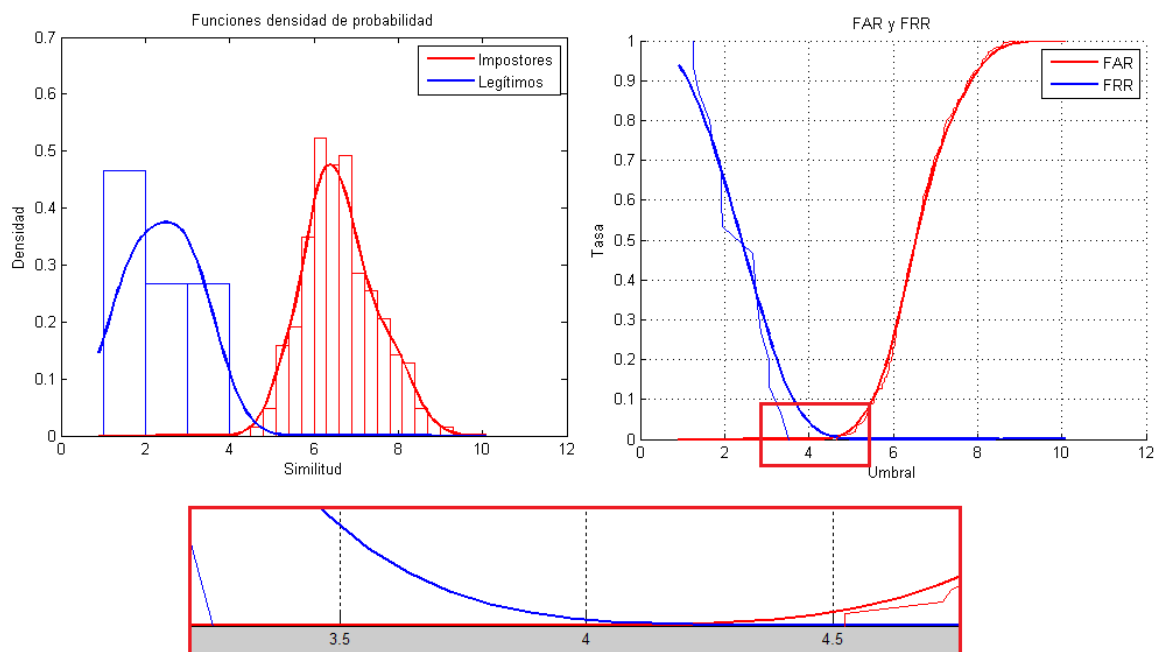


Figura 10.8. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en LDA (BD:YALE).

Obsérvese nuevamente que aunque las estimaciones se crucen, se puede trazar un umbral con los datos empíricos que separe perfectamente las curvas de la FAR y la FRR. Nótese que para el mejor caso de LDA la holgura dentro de la cual se puede mover el umbral asegurando tasas de error nulas para el set de validación utilizado es mucho mayor

que para el mejor caso de PCA. Obsérvese también lo pequeña que es la zona donde se cruzan la FAR y la FRR de las estimaciones.

Wavelets de Gabor

Para evaluar el sistema basado en *wavelets* de Gabor se optó por utilizar 3 tamaños distintos de mallas de puntos: una malla pequeña (4x6 puntos), una mediana (6x8 puntos) y una grande (10x10 puntos). El tamaño de la malla grande es el tamaño máximo permitido por el programa diseñado.

El procedimiento de evaluación fue comparar primero los tres tamaños de la malla de puntos para cada pre-procesado posible obteniendo 3 mejores curvas, que fueron luego comparadas para obtener a configuración representante del sistema basado en *wavelets* de Gabor.

Dicho esto, para ecualización de histograma, las ROC y DET se observan en la Figura 10.9; para normalización de varianza, se observan en la Figura 10.10; y para pre-procesado mixto, las curvas se observan en la Figura 10.11.

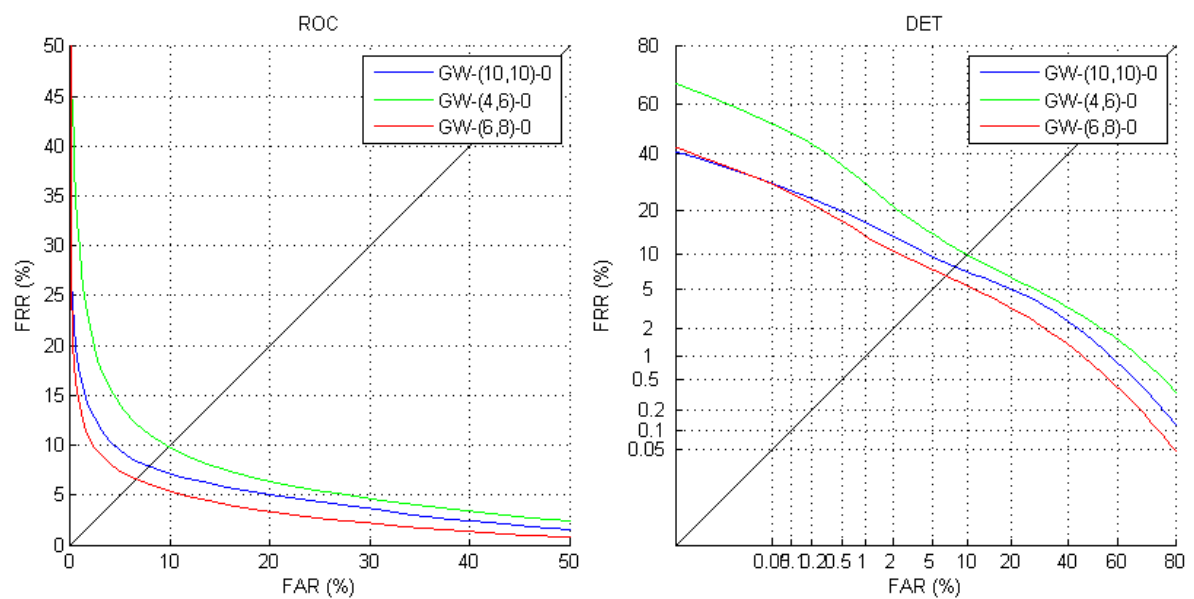


Figura 10.9. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado: Ecualización de histograma) (BD:YALE).

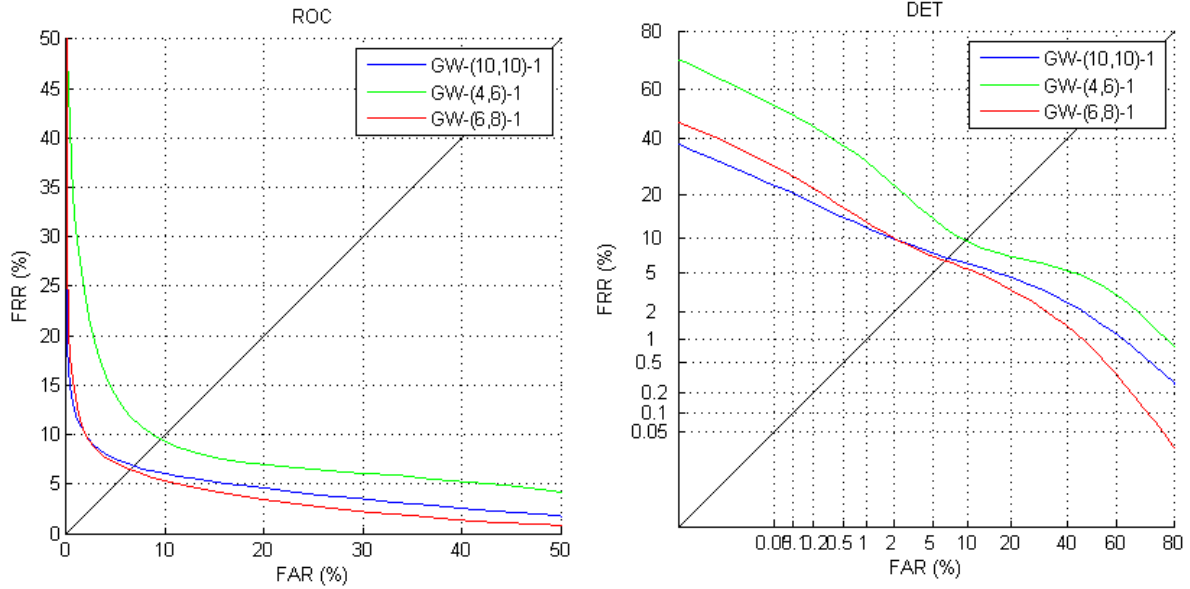


Figura 10.10. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado: Normalización de varianza). (BD:YALE).

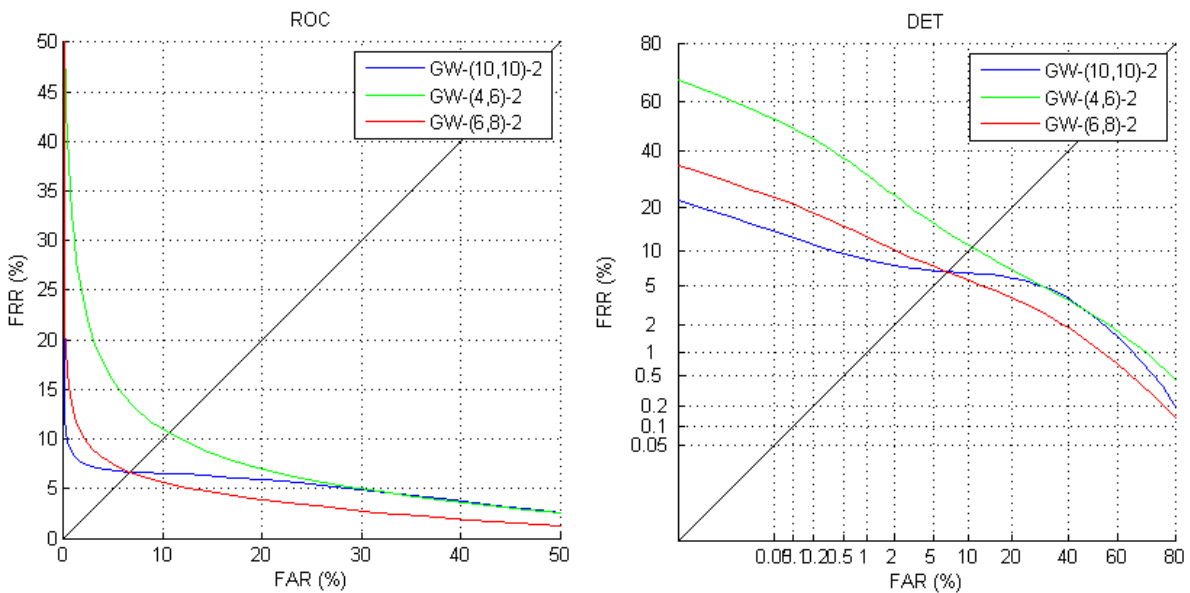


Figura 10.11. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado: Ecualización de histograma y normalización de varianza)(BD:YALE).

Se puede observar que la malla con mejor EER en todos los casos es la de la malla 6x8, por lo que en la Figura 10.12 se comparan todos los tipos de pre-procesado para dicha malla. Excepcionalmente, se muestran también las curvas para la malla de puntos de 10x10 con pre-procesado mixto. Esta decisión se debe a sus buenas tasas de falso rechazo para bajas tasas de falsa aceptación.

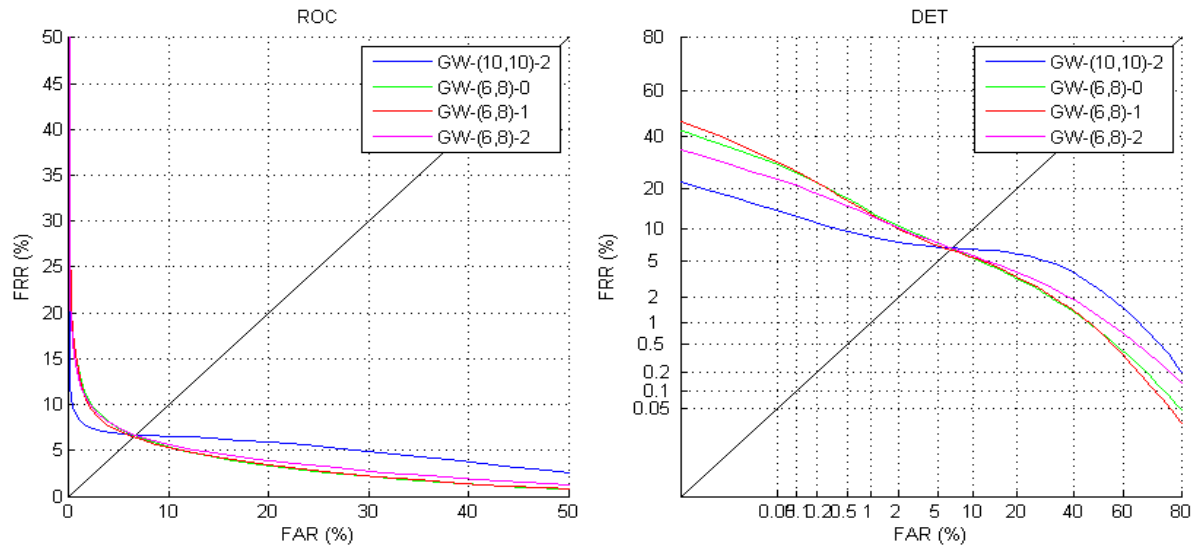


Figura 10.12. Curvas ROC y DET para todos los posibles pre-procesados de la malla de puntos de 6x8 y para el pre-procesado mixto de la malla de 10x10 (BD:YALE).

Observando de los resultados mostrados en la Figura 10.12, se hace es difícil elegir una configuración representante. Si se desearan bajas tasas de falso rechazo cuando son preferibles tasas de falsa aceptación bajas (alta seguridad), entonces lo ideal sería escoger el de la malla de 10x10. Si por lo contrario se deseara un sistema que tenga bajas tasas de falsa aceptación cuando son preferibles tasas de falso rechazo bajas, entonces habría que escoger la malla de 6x8 puntos con normalización de varianza de pre-procesado. No obstante, para comparar el sistema basado en *wavelets* de Gabor con los sistemas basados en PCA y LDA, se escogió la malla de 6x8 puntos con pre-procesado mixto ya que supone un término medio de rendimiento. Las densidades de probabilidad de legítimos e impostores y las FAR y FRR en función del umbral para dicha configuración se muestran en la Figura 10.13. Nótese que para este sistema las curvas de FAR y FRR empíricas sí se solapan, no pudiéndose trazar un umbral que separe perfectamente ambas curvas.

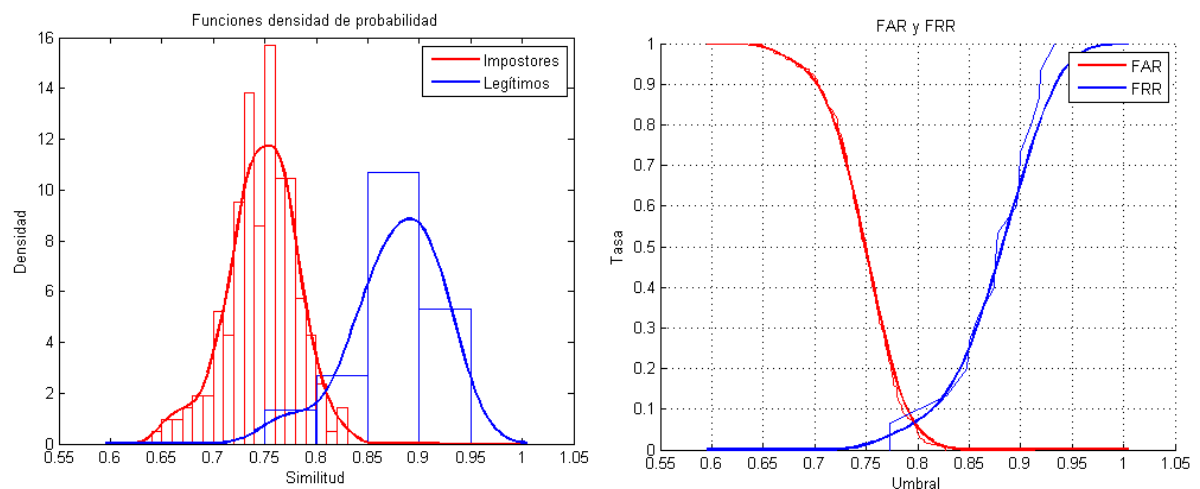


Figura 10.13. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en *wavelets* de Gabor (BD:YALE).

Comparación entre los tres sistemas

En la Figura 10.14 se pueden observar las ROC y DET de las configuraciones representativas para cada sistema de reconocimiento facial, donde se puede observar que el sistema con mejor rendimiento resultante es el que utiliza LDA como técnica de extracción de características. El segundo sistema con mejor rendimiento resultó ser el sistema basado en PCA, y el último el sistema basado en *wavelets* de Gabor.

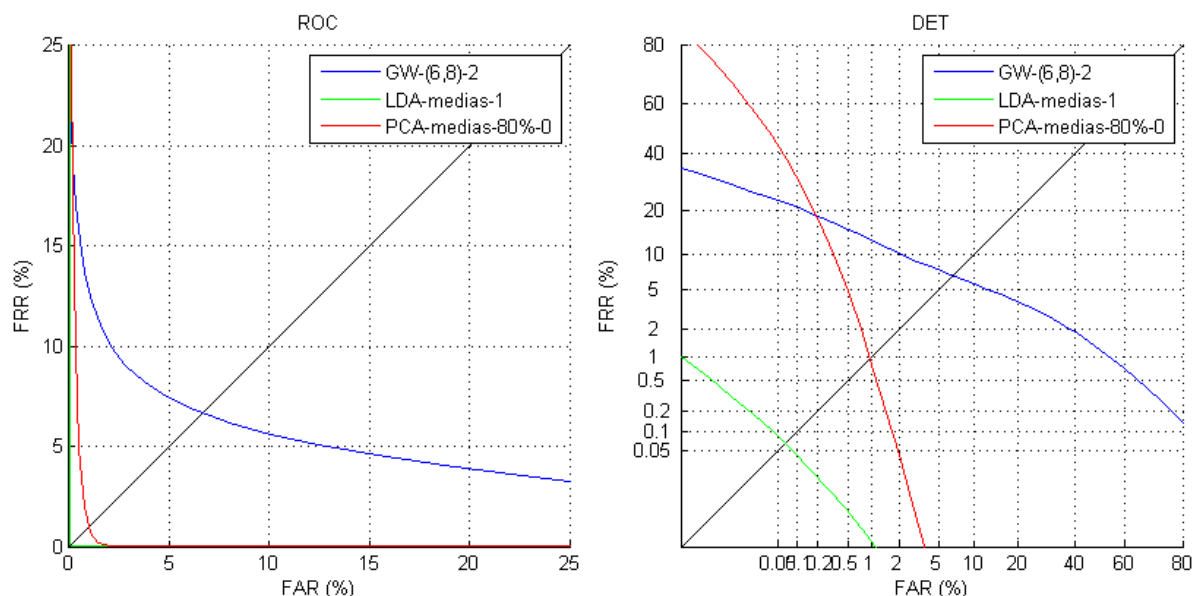


Figura 10.14. Comparación general de los sistemas basados en PCA, LDA, y *wavelets* de Gabor (BD: YALE).

10.2. Base de datos del Centro Universitário da FEI

Para el conjunto de entrenamiento se utilizaron las 5 imágenes más frontales de 15 individuos de la base de datos. No obstante, varias de las imágenes de entrenamiento utilizadas resultaron con la cabeza un poco girada.

Para la segmentación de rostros, se utilizó detección automática con los porcentajes de corte adicional indicados en la Tabla 10.2.

Porcentajes de corte adicional (%)	
Porcentaje lateral	-10
Porcentaje superior	0
Porcentaje inferior	5

Tabla 10.2. Porcentajes de corte adicional utilizados para la segmentación de rostros de la base de datos del Centro Universitário da FEI.

Un ejemplo del corte utilizado con la detección automática se puede observar en la Figura 10.15.



Figura 10.15. Tipo de corte utilizado para segmentar la imágenes de la base de datos del Centro Universitário da FEI.

Existen 3 diferencias fundamentales entre este conjunto de entrenamiento y el utilizado para la base de datos de *Yale University* que se consideran desfavorables para la clasificación:

1. Se utilizan solamente 5 imágenes de entrenamiento en lugar de 10.
2. Entre las imágenes de entrenamiento existen distintos ángulos de rotación de la cabeza.
3. Se utiliza la detección automática, que conlleva a una segmentación menos uniforme de los rostros.

PCA

Se siguió el mismo procedimiento utilizado para la base de datos de *Yale University*: Se comparó primero el sistema a resolución máxima con todas las posibles combinaciones de tipo de umbral y pre-procesado (Figura 10.16).

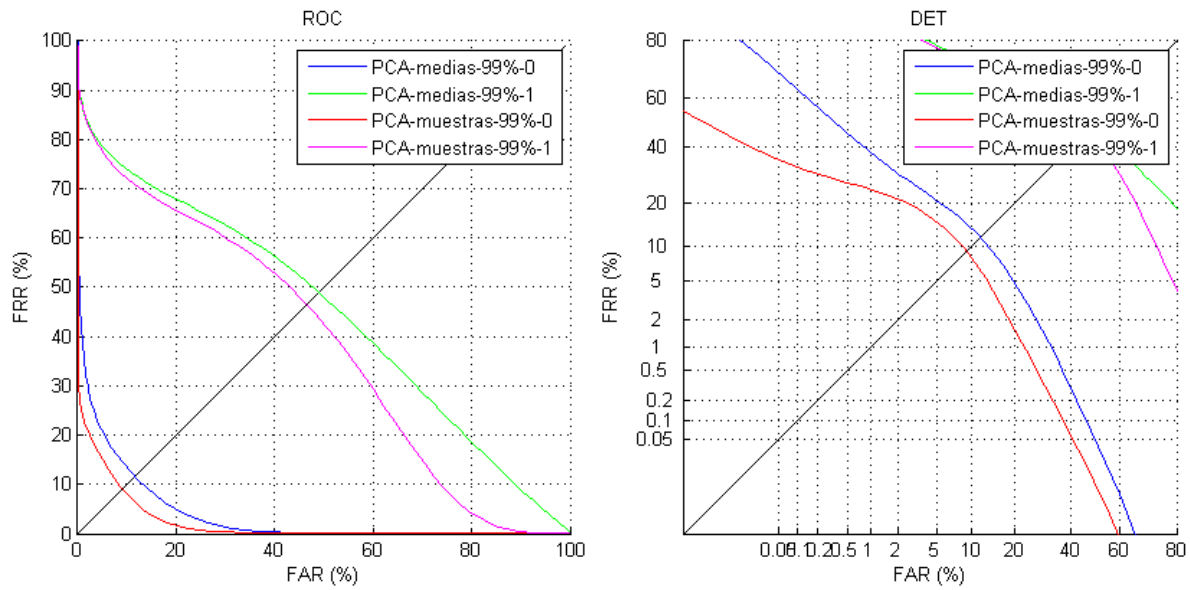


Figura 10.16. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral a una resolución del 100% (BD:FEI).

A diferencia de la base de datos anterior, la configuración que consiguió una mejor precisión fue la del umbral por muestras con ecualización de histograma de pre-procesado. Además, para esta base de datos (y las condiciones especificadas) es más notable la influencia positiva del pre-procesado en el reconocimiento.

A continuación, se presenta en la Figura 10.17 las ROC y DET de la mejor configuración resultante de la primera comparación a las resoluciones mínimas del 20%, 50% y 75%. Nótese que a medida de que disminuye la resolución empeora el rendimiento y se obtienen curvas más deformes.

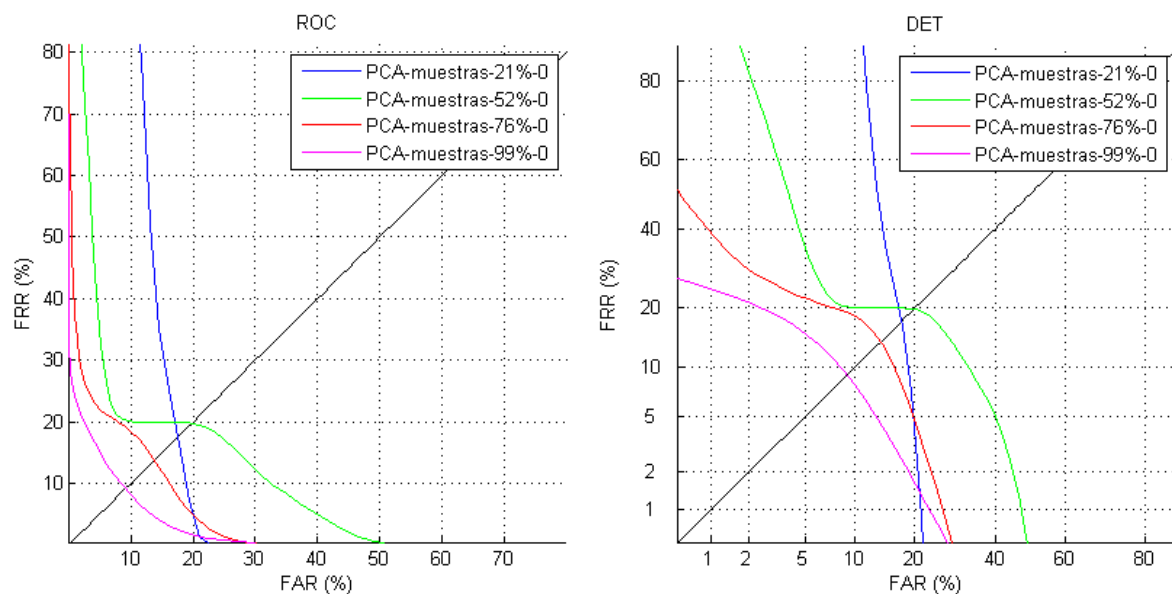


Figura 10.17. Curvas ROC y DET para distintas resoluciones de un umbral por muestras con ecualización de histograma de pre-procesado (BD:FEI).

Según los resultados obtenidos en la Figura 10.17, es evidente que la distribución de los datos con baja resolución no se aproxima en absolutamente nada a una distribución normal ni tampoco supone una mejora en la clasificación. En consecuencia, no se llevó a cabo una evaluación a baja resolución para esta base de datos.

Dicho esto, la configuración representante del PCA para esta base de datos resultó ser la del tipo de umbral por muestras con ecualización de histograma a una resolución del 100%. Para dicha configuración se muestran las densidades de probabilidad de legítimos e impostores y las FAR y FRR en función del umbral en la Figura 10.18.

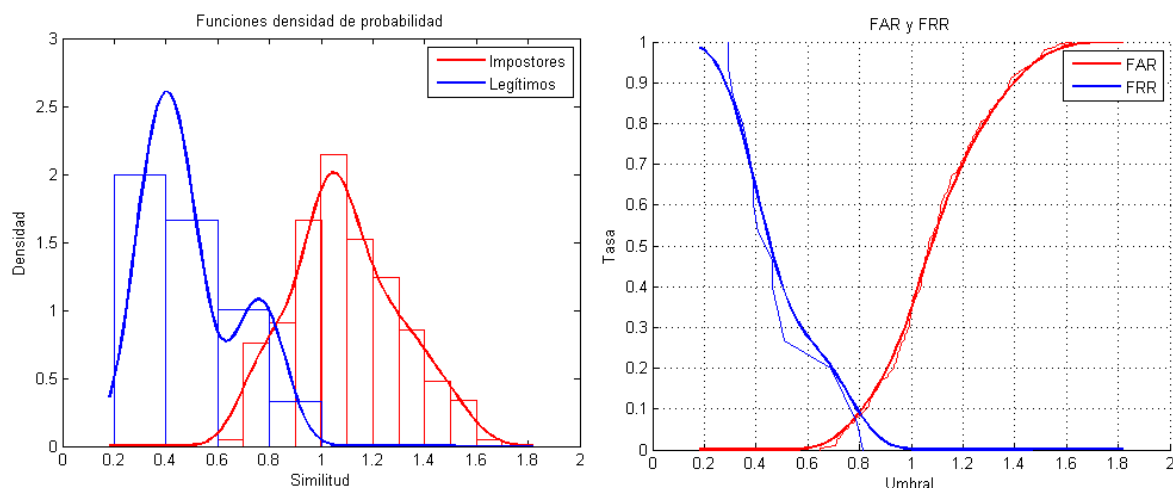


Figura 10.18. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en PCA (BD:FEI).

LDA

Al igual que para la base de datos de *Yale University*, se compararon todas las configuraciones posibles (Figura 10.19).

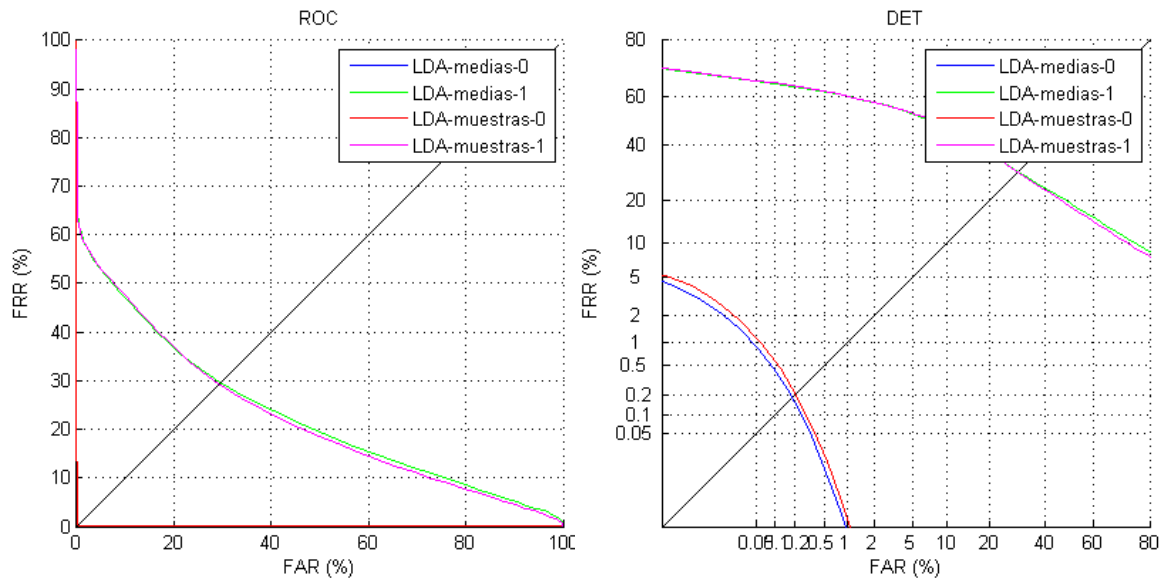


Figura 10.19. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral para un sistema basado en LDA. No es posible representar de manera visible las cuatro ROC en una sola gráfica (BD:FEI).

Es importante destacar cuenta que en la Figura 10.19 (a la izquierda) se representan todas las ROC medidas. Sin embargo, la mejora que supone aplicarle en este caso una ecualización de histograma a las imágenes no permite que sean comparables con las ROC sin pre-procesado. En casos como este es cuando la DET cumple mejor con su objetivo. Nótese que las curvas se encuentra mucho mejor separadas graficando sus DET en lugar de las ROC.

La configuración representante del LDA para esta base de datos resulta la de umbral por medias con ecualización de histograma como pre-procesado. Sus densidades de probabilidad de legítimos e impostores, y sus FAR y FRR en función del umbral se muestran en la Figura 10.20. Obsérvese que las curvas empíricas de la FAR y la FRR quedan lo suficientemente separadas para trazar un umbral sin tasas de error. Obsérvese también lo pequeña que es el área de intersección entre las curvas FAR y FRR de las estimaciones de la distribución.

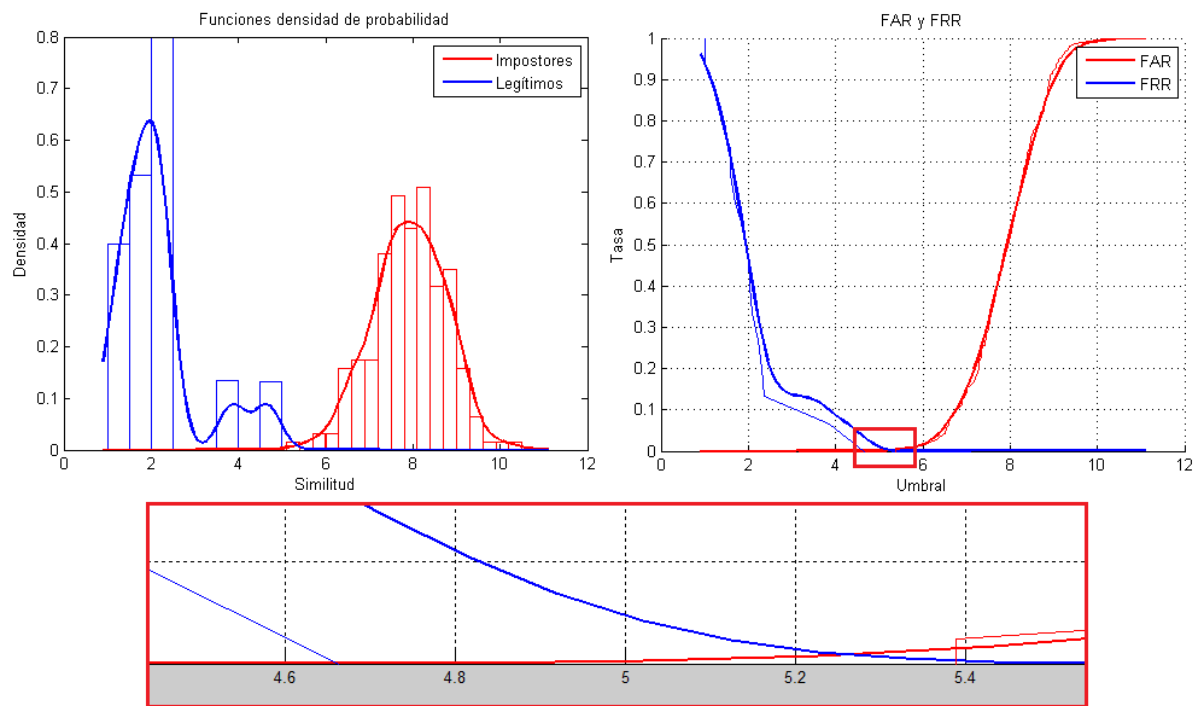


Figura 10.20. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en LDA (BD:FEI).

Wavelets de Gabor

Siguiendo el mismo procedimiento de evaluación de configuraciones de *wavelets* de Gabor de la base de datos anterior, las ROC y DET de los distintos pre-procesados para las mallas de 4x6, 6x8, y 10x10 se muestran en la Figura 10.21, Figura 10.22 y Figura 10.23.

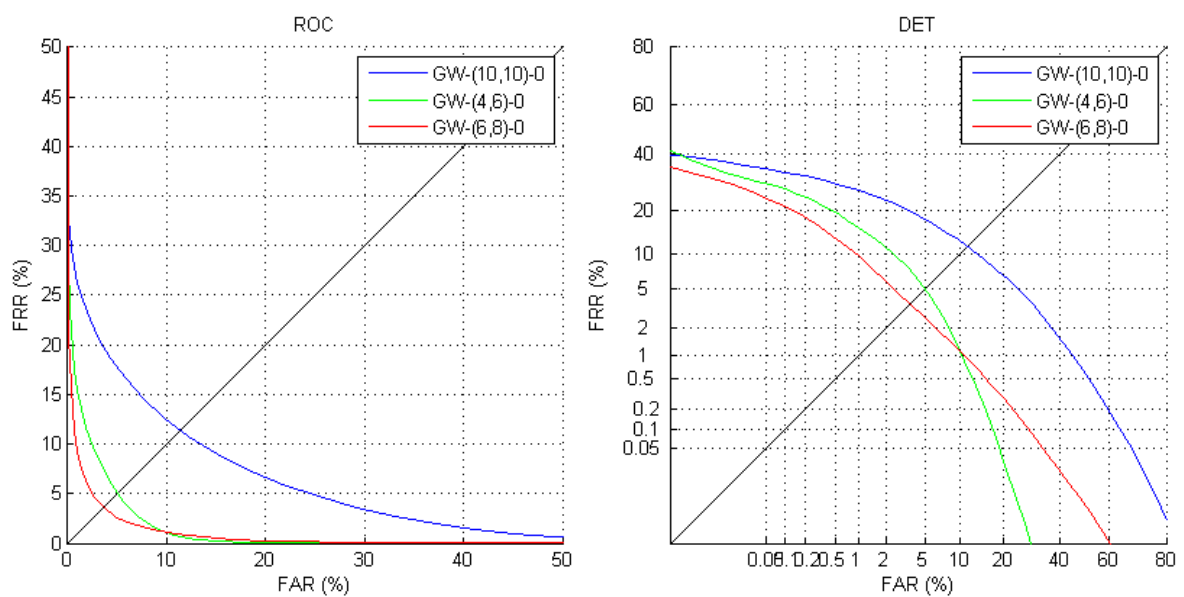


Figura 10.21. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Ecuación de histograma) (BD:FEI).

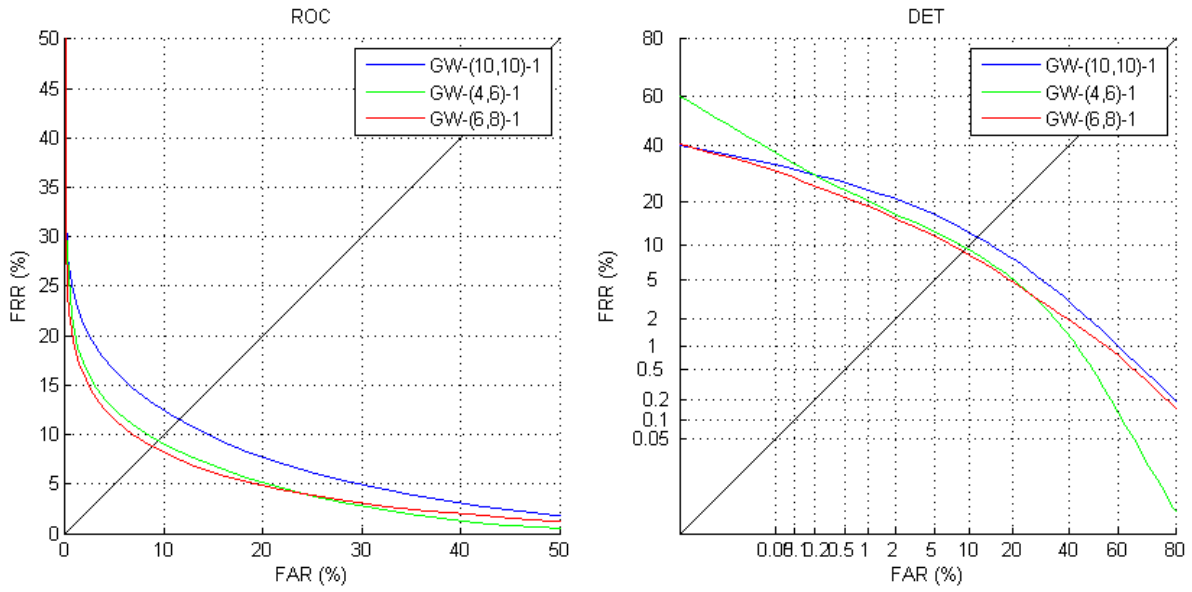


Figura 10.22. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Normalización de varianza) (BD:FEI).

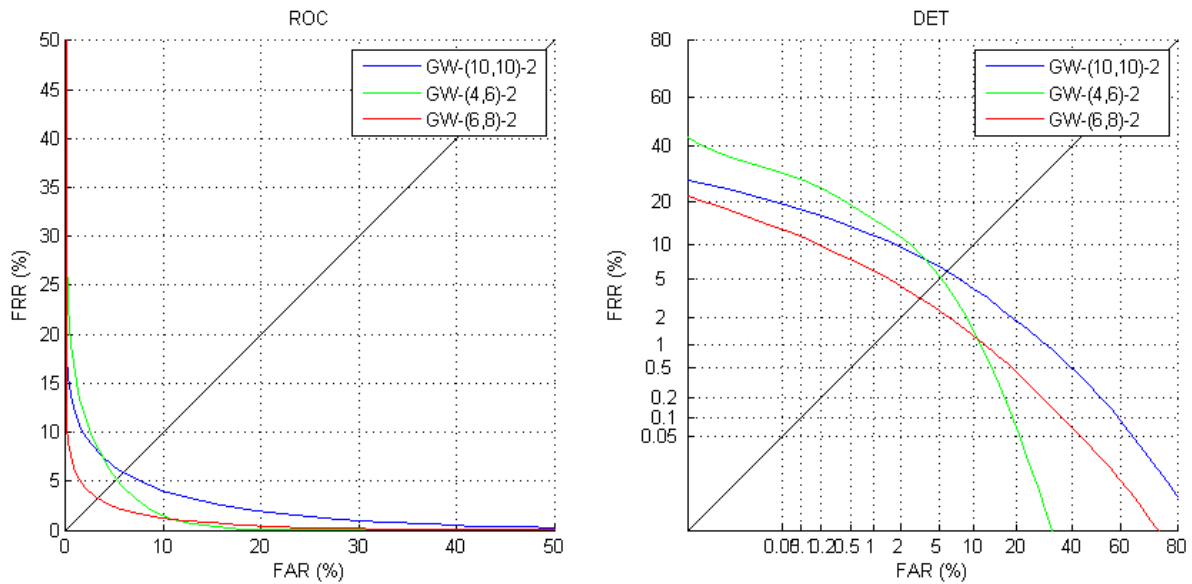


Figura 10.23. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado mixto) (BD:FEI).

Al igual que para la base de datos de *Yale University*, los mejores resultados se obtuvieron con la malla de puntos de 6x8. Por lo tanto, se presenta en la Figura 10.24 la comparación entre todos los posibles pre-procesados para dicha malla.

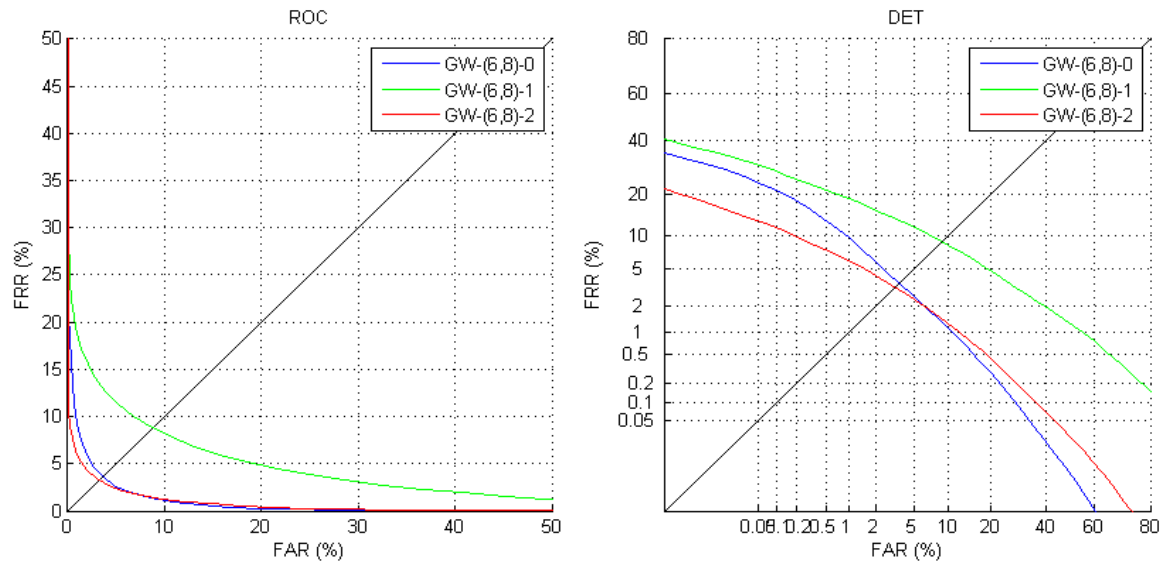


Figura 10.24. Curvas ROC y DET para los posibles pre-procesados de la malla de puntos de 6x8 (BD:FEI).

En función de los resultados obtenidos de la Figura 10.24, se elige como representante de las *wavelets* de Gabor la configuración de la malla de puntos de 6x8 con pre-procesado mixto. En la Figura 10.25 se muestran sus densidades de probabilidad de individuos legítimos e impostores y sus curvas FAR y FRR en función del umbral. Nótese que aunque sea por poco, las curvas de FAR y FRR en función del umbral obtenidas a partir de los datos empíricos están separadas al igual que para el sistema basado en LDA y a diferencia del sistema basado en PCA.

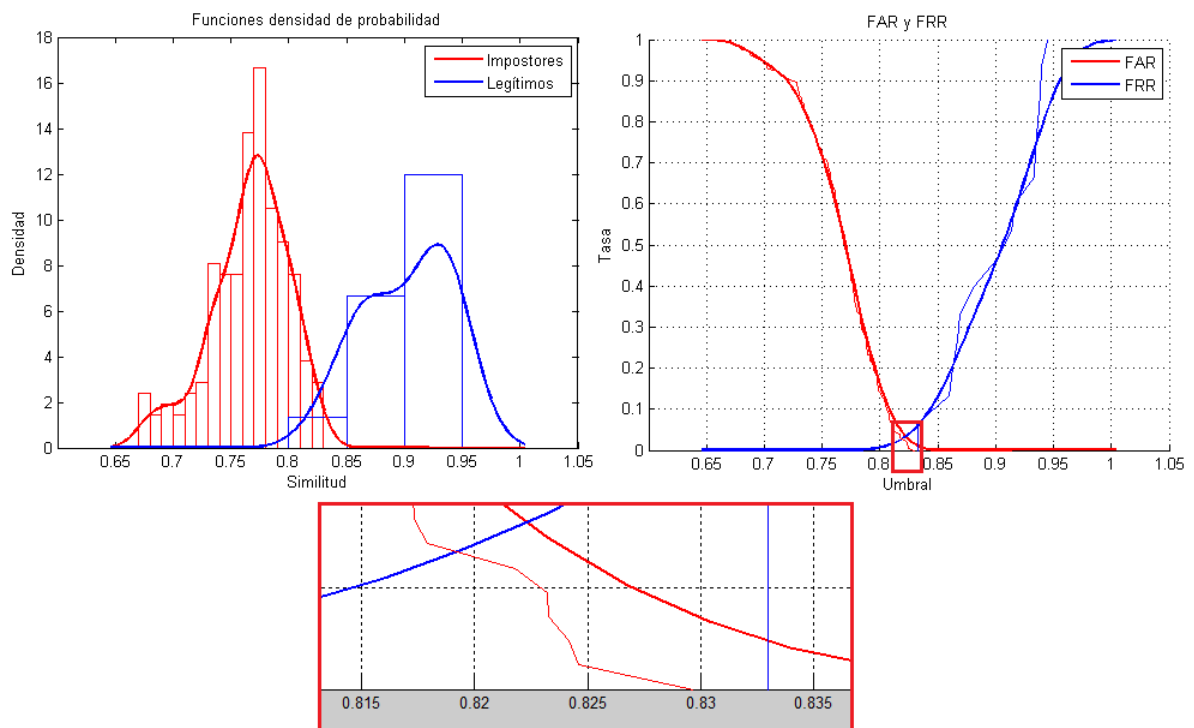


Figura 10.25. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en wavelets de Gabor (BD:FEI).

Comparación entre los tres sistemas

La comparación entre los representantes de cada sistema se presenta en la Figura 10.26. Nótese que el sistema de reconocimiento facial basado en LDA vuelve a ser el que tiene mejor rendimiento. Sin embargo, para esta base de datos (y con las condiciones especificadas) es preferible utilizar un sistema basado en *wavelets* de Gabor que un sistema basado en PCA.

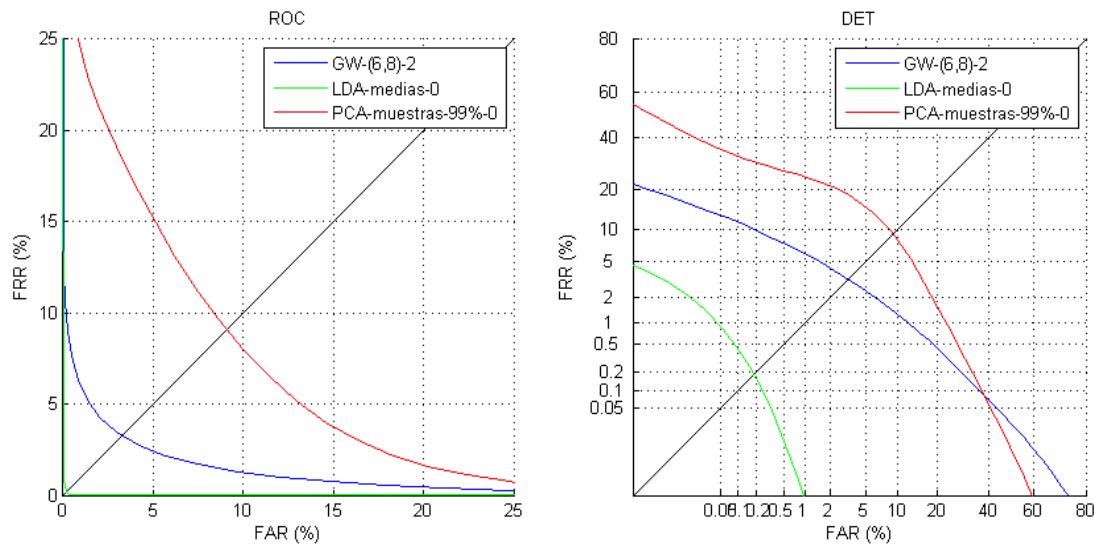


Figura 10.26. Comparación general de los sistemas basados en PCA, LDA, y wavelets de Gabor (BD:FEI).

10.3. Base de datos personal no-controlada

La diferencia principal que posee esta base de datos frente a las dos anteriores es que las condiciones de adquisición son muy variadas y generalmente muy desfavorables. Se utilizó la misma segmentación manual y con los mismos porcentajes de corte que para la base de datos de *Yale University* (Tabla 10.1). El número de imágenes de entrenamiento por cada individuo fue de 5.

Un ejemplo de una de las imágenes que forman a la base de datos se puede observar en la Figura 10.27.



Figura 10.27. Ejemplo de una de las imágenes de la base de datos personal no-controlada.

PCA

En la Figura 10.28 se muestra la comparación entre todas las configuraciones posibles para una resolución máxima.

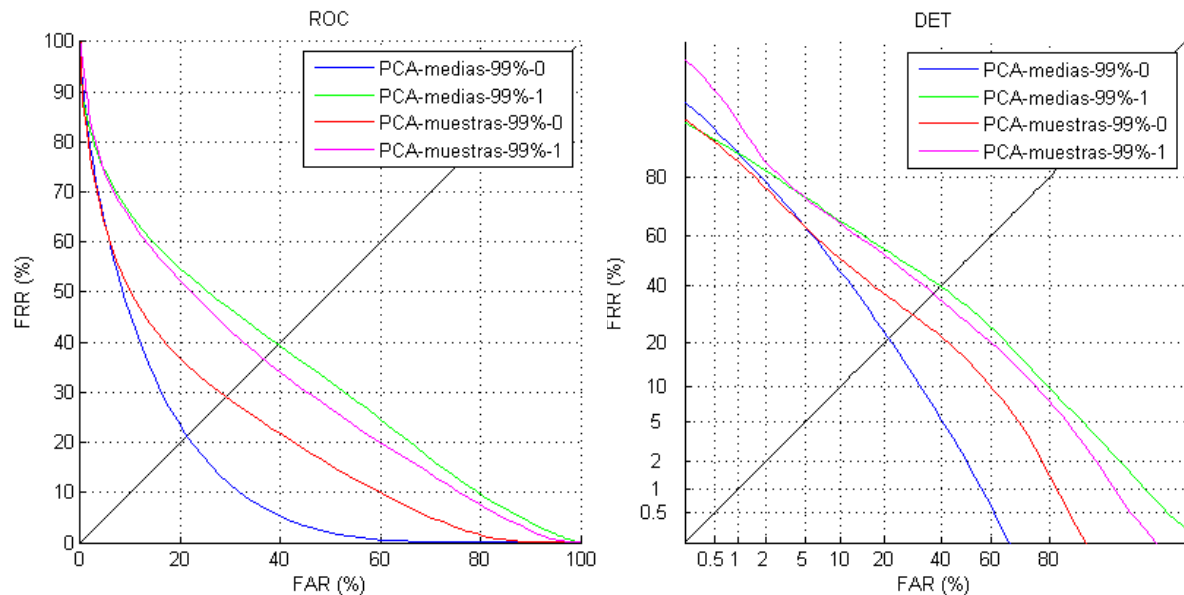


Figura 10.28. Curvas ROC y DET de todas las configuraciones posibles para una resolución máxima (BD:PERSONAL).

Al igual que en la base de datos de *Yale University*, la configuración con mayor precisión es la de la configuración del umbral por medias con ecualización de histograma. Se utilizó dicha configuración para hacer la comparación de resoluciones (Figura 10.29). Nótese que al ser una base de datos con condiciones de adquisición no-controladas, las curvas ROC demuestran poco rendimiento en comparación con las otras bases de datos.

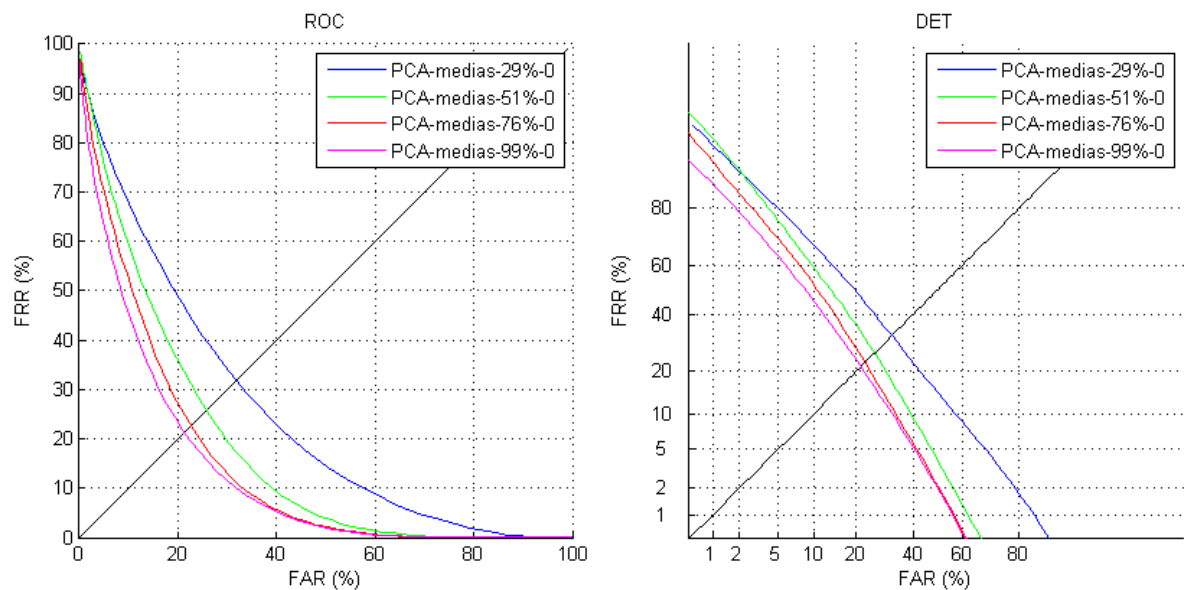


Figura 10.29. Curvas ROC y DET para distintas resoluciones de una configuración de umbral por muestras y ecualización de histograma como pre-procesado (BD:PERSONAL).

Según se puede observar en la Figura 10.29, y al igual que en la base de datos del *Centro Universitário da FEI*, a medida de que disminuye la resolución, disminuye la precisión del sistema.

Se eligió la como representante la configuración del umbral por medias con ecualización de histograma y a resolución máxima. Sus densidades de probabilidad de legítimos e impostores y curvas de FAR y FRR en función del umbral se observan en la Figura 10.30.

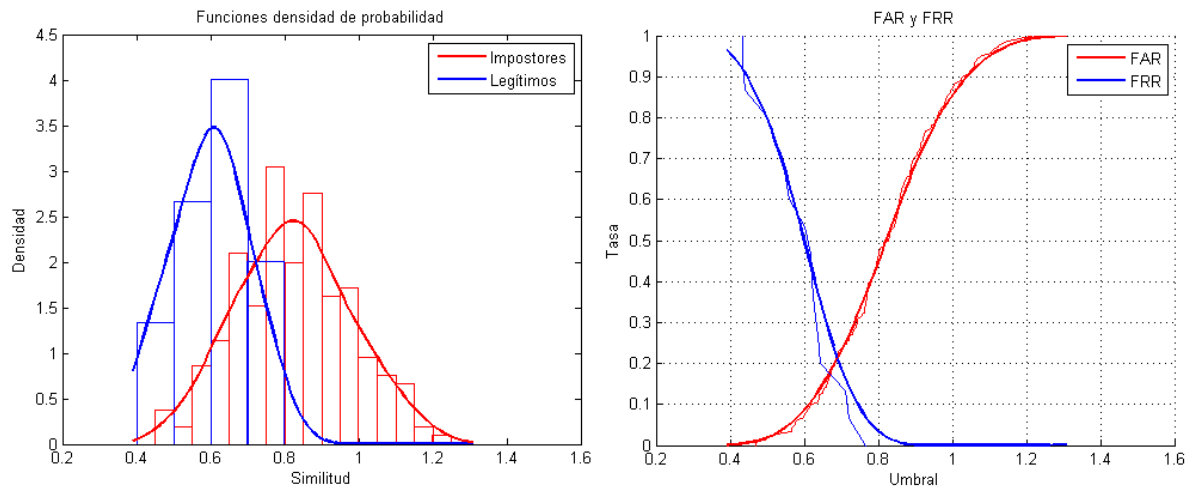


Figura 10.30. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en PCA (BD:PERSONAL).

LDA

En la Figura 10.31 se muestran todas las combinaciones posibles configuraciones. De ella se puede deducir que la configuración de LDA con mejor rendimiento es la del umbral por medias con ecualización de histograma (se escoge como representante).

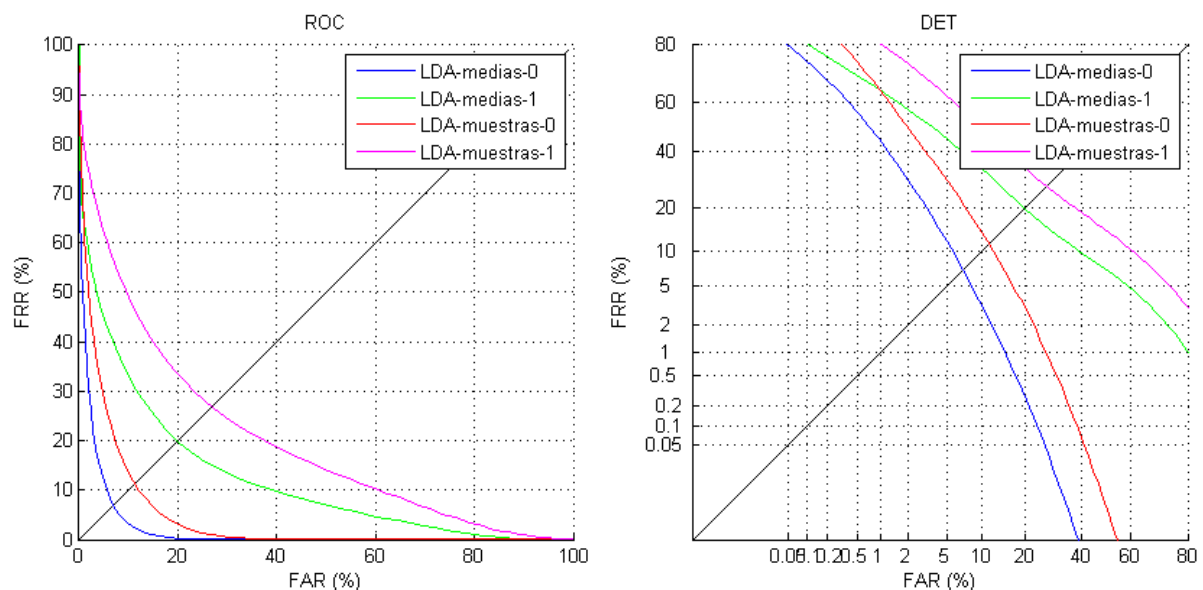


Figura 10.31. Curvas ROC y DET para todas las configuraciones posibles del sistema basado en LDA (DB:PERSONAL).

Otro detalle que no debe pasar desapercibido es la mejora del rendimiento del sistema al haber introducido un pre-procesado adecuado sobre imágenes con condiciones de captura no-controladas (en este caso, ecualización de histograma).

Las densidades de probabilidad de individuos y las curvas FAR y FRR en función del umbral se pueden observar en la Figura 10.32.

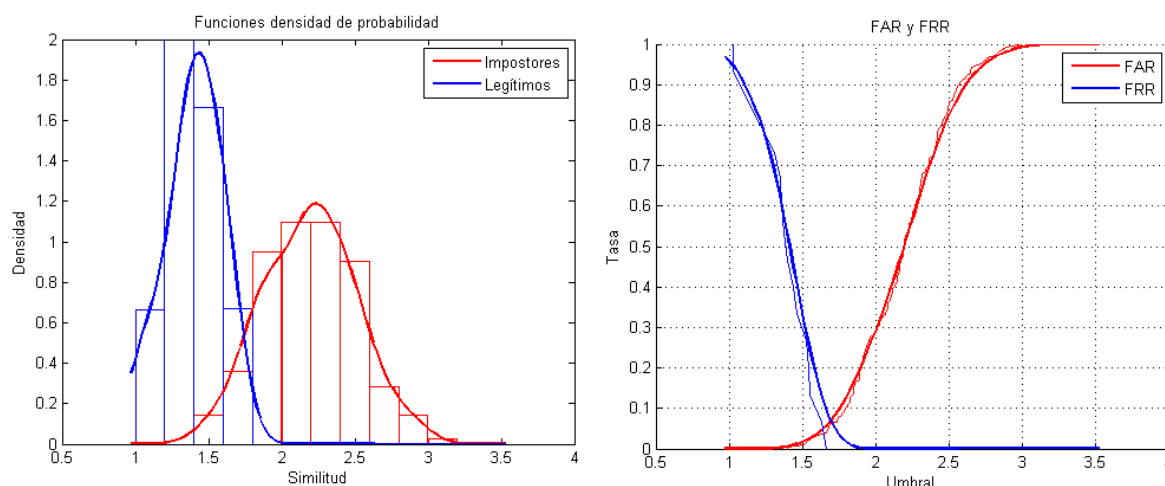


Figura 10.32. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en LDA (BD:PERSONAL).

Wavelets de Gabor

Se realizaron las mismas comparaciones que para las bases de datos anteriores. En la Figura 10.33 se muestran las mismas 3 mallas de puntos utilizadas en las bases de datos anteriores con ecualización de histograma, en la Figura 10.34 con normalización de varianza, y en la Figura 10.35 con pre-procesado mixto.

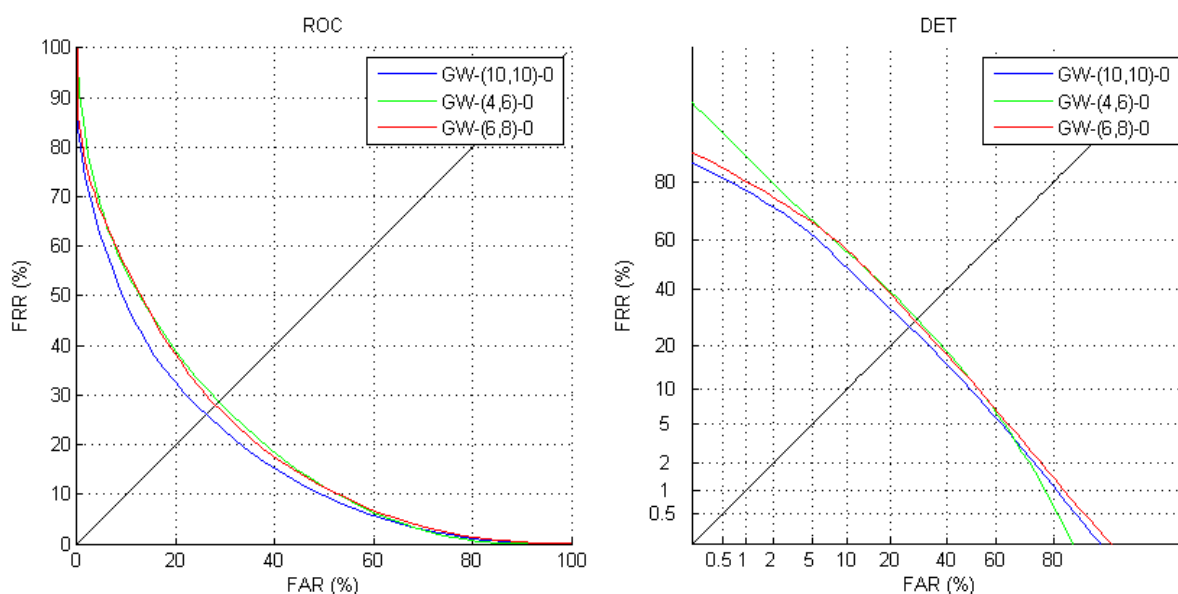


Figura 10.33. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Ecualización de histograma) (BD:PERSONAL).

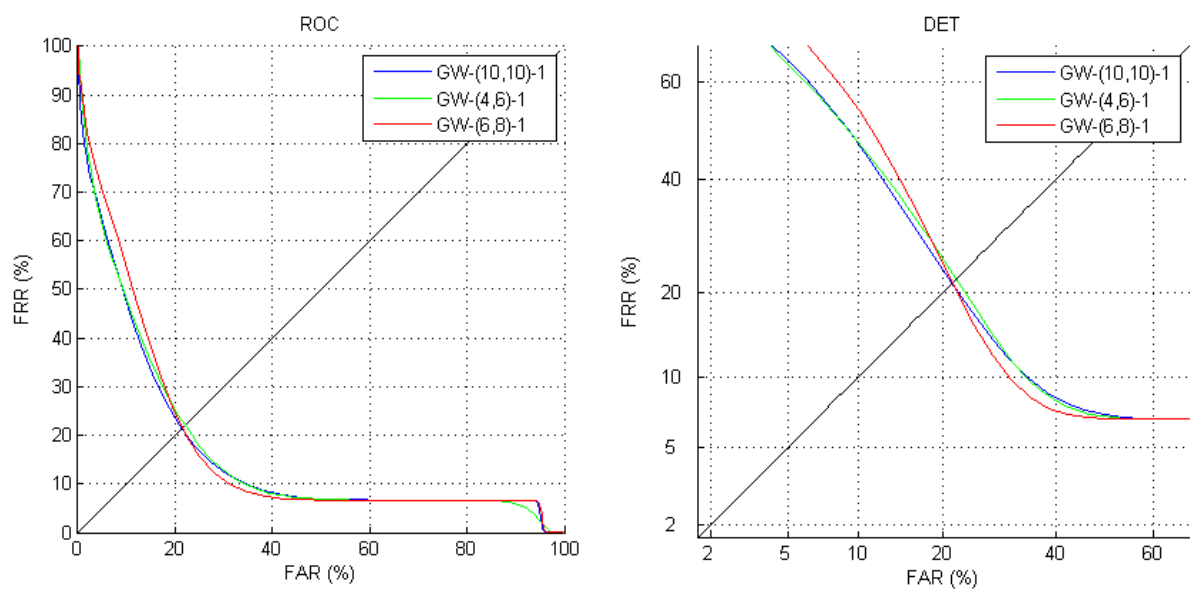


Figura 10.34. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Normalización de varianza) (BD:PERSONAL).

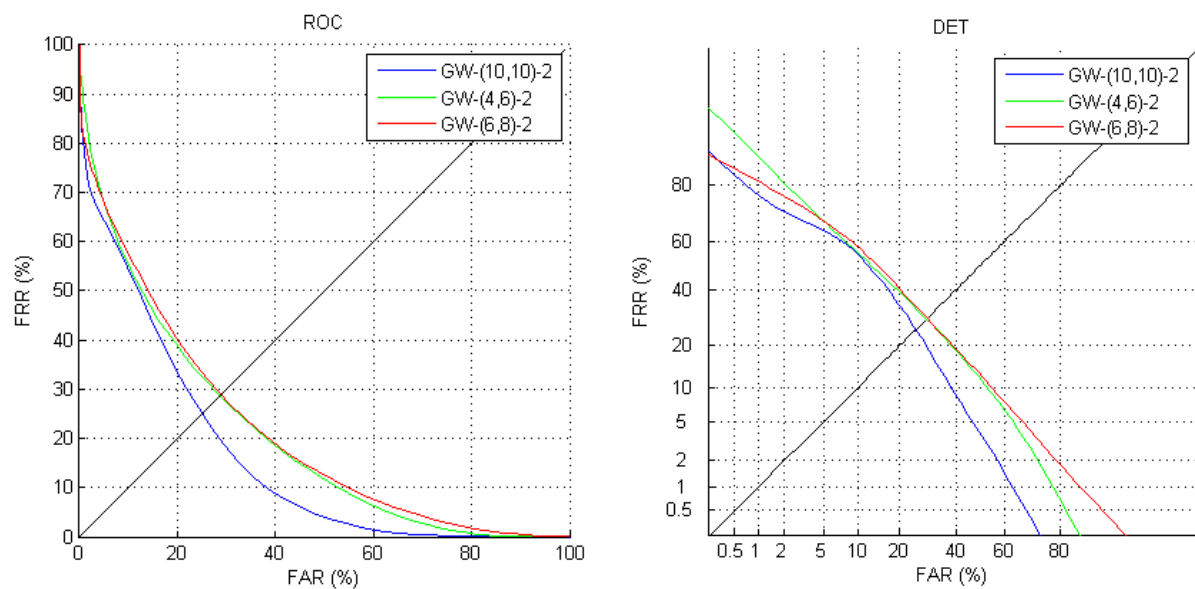


Figura 10.35. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado mixto) (BD:PERSONAL).

A diferencia de las bases de datos anteriores, la malla con más rendimiento resultó ser la de 10x10, en lugar de la de 6x8. Por lo tanto, en la Figura 10.36 se muestran las curvas ROC y DET para todos los pre-procesados.

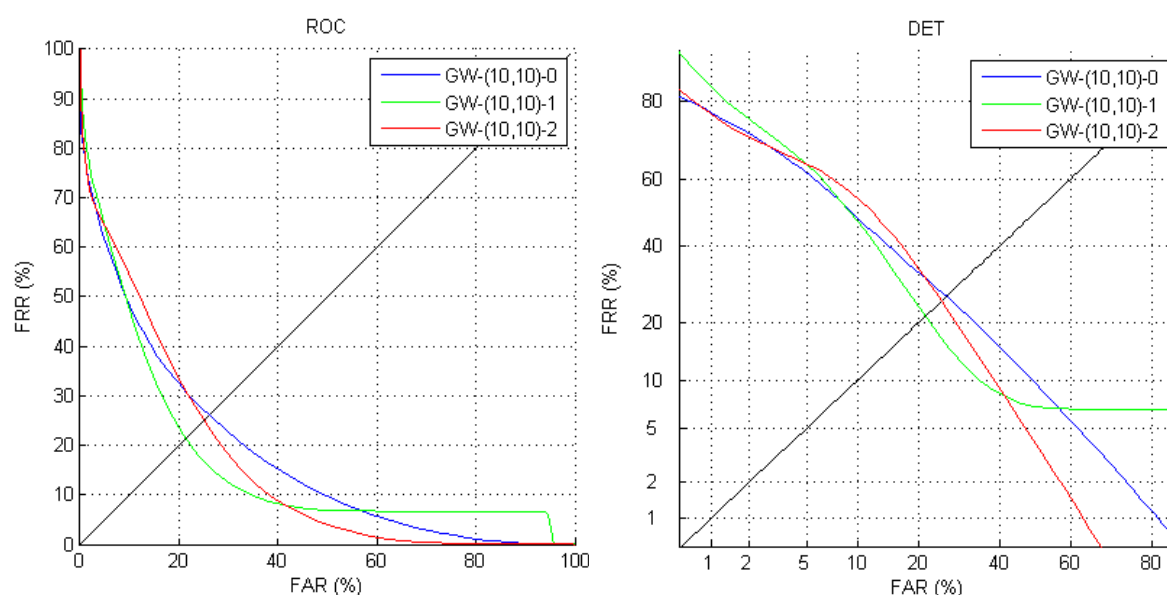


Figura 10.36. Curvas ROC y DET de todos los pre-procesados posibles para la malla de puntos de 10x10 (BD: PERSONAL).

A pesar de que la configuración con menor EER resultó ser la de normalización de varianza, se descartó como candidata a representante por las altas tasas de falsa aceptación que son necesarias para reducir el falso rechazo por debajo del 5%. Como consecuencia, se escogió como representante la configuración de pre-procesado mixto.

Las curvas de FAR y FRR en función del umbral para la configuración representante escogida pueden observarse en la Figura 10.37.

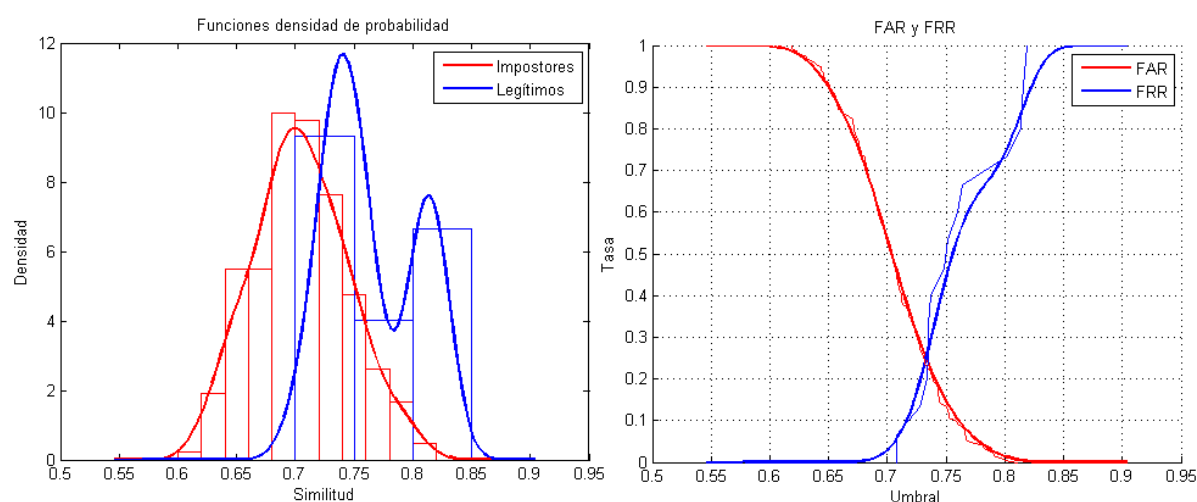


Figura 10.37. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en wavelets de Gabor (BD:PERSONAL).

Comparación entre los tres sistemas

En la Figura 10.38 se puede observar la comparación final de todos los sistemas para la base de datos personal no-controlada.

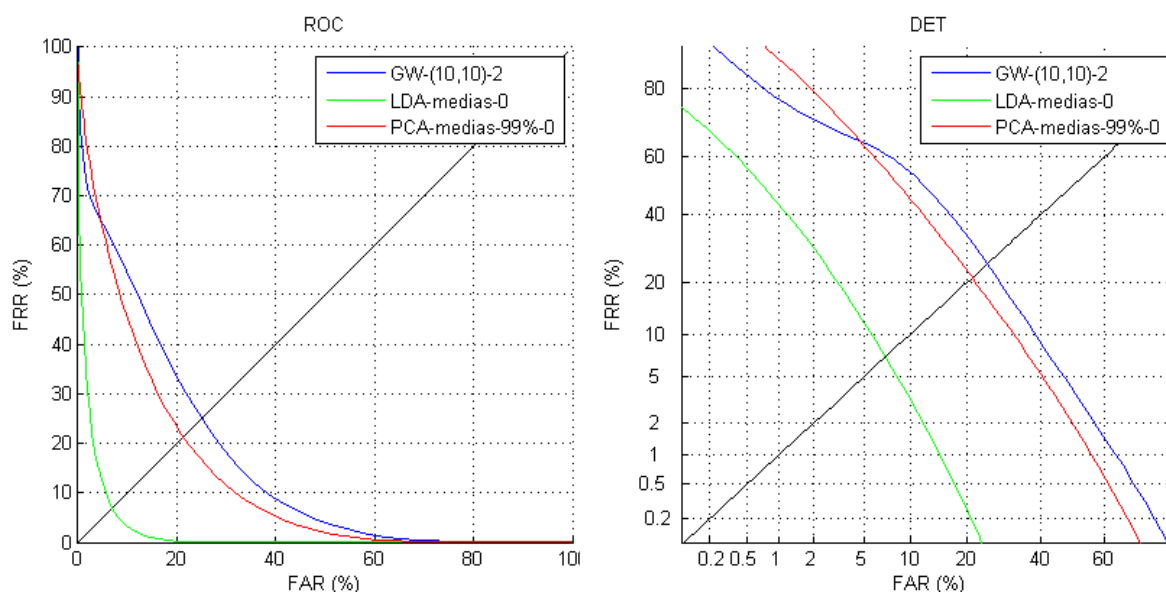


Figura 10.38. Comparación general de los sistemas basados en PCA, LDA, y wavelets de Gabor (BD:PERSONAL).

De la figura se deduce que el sistema con mejor rendimiento vuelve a ser el sistema basado en LDA, seguido por el de PCA, y por último el de *wavelets* de Gabor.

10.4. Comparación entre el modo de verificación tradicional y el modo de verificación híbrido propuesto

Como se comentó en el capítulo 9.13, el objetivo perseguido al utilizar el modo de verificación híbrido propuesto es reducir las tasas de falsa aceptación de un sistema de clasificación trabajando en modo verificación. En consecuencia, para comprobar los efectos de utilizar el modo de verificación híbrido en lugar del modo de verificación tradicional, se decidió fijar un umbral para el cual se asegure una tasa nula de falso rechazo sobre los datos empíricos en cada una de las configuraciones representantes de cada base de datos.

De esta manera se pretendía responder a la siguiente pregunta: ¿Es posible, para los datos obtenidos de forma empírica, poner al mismo nivel el rendimiento de las configuraciones en cuyas evaluaciones no se consigue separar las curvas FAR y FRR en función del umbral y el rendimiento las configuraciones en las que sí?

Antes de presentar las tablas de resultados del modo de verificación híbrido, se llamará por simplicidad **umbral de comodidad** a aquel umbral que asegure una tasa de falso rechazo nula. El nombre escogido se debe al hecho de que si se asegura una tasa nula de falso rechazo, teóricamente no debería haber ningún problema en la práctica para que

un individuo legítimo acceda al sistema, independientemente de la tasa de falsa aceptación (es un sistema de acceso cómodo porque ningún legítimo tiene que pasar por la molestia de ser rechazado).

Los umbrales de comodidad fueron obtenidos a partir de las gráficas de FAR y FRR en función del umbral para las configuraciones representantes de cada base de datos (Figura 10.6, Figura 10.8, Figura 10.13, Figura 10.18, Figura 10.20, Figura 10.25, Figura 10.30, Figura 10.32, y Figura 10.37), y utilizando cursores en MATLAB.

En la Tabla 10.3 se muestran las tasas de falso rechazo y de falsa aceptación obtenidas utilizando ambos modos de verificación con el umbral de comodidad para las curvas obtenidas con los datos empíricos.

	Configuración	Umbral	Tradicional		Híbrido	
			FRR (%)	FAR (%)	FRR (%)	FAR (%)
YALE	PCA-medias-80%-0	0,340	0	0	0	0
	LDA-medias-1	3,900	0	0	0	0
	GW-(6,8)-2	0,770	0	24,7619	0	0
FEI	PCA-muestras-99%-0	0,815	0	9,0476	6,6667	0,4762
	LDA-medias-0	5,000	0	0	0	0
	GW-(6,8)-2	0,830	0	0	0	0
PERSONAL	PCA-medias-99%-0	0,780	0	39,0476	33,3333	2,3810
	LDA-medias-0	1,670	0	4,2857	0	0
	GW-(10,10)-2	0,700	0	53,8095	40,0000	2,8571

Tabla 10.3. Comparación entre el modo de verificación tradicional y el híbrido. Las tasas fueron calculadas sobre los datos obtenidos de forma empírica, y se fijó el umbral de comodidad basándose en los datos obtenidos de forma empírica.

Las configuraciones de la Tabla 10.3 resaltadas en color rojo representan aquellas configuraciones para las cuales fue posible separar las curvas FAR y FRR en función del umbral. Consecuentemente, para estas configuraciones las tasas de error resultaron nulas. Obsérvese que utilizando el modo de verificación híbrido para estas configuraciones, los resultados son los mismos.

Son las configuraciones resaltadas en color verde las que dan una respuesta afirmativa a la pregunta planteada. Nótese que utilizando el modo de verificación tradicional sobre estas configuraciones se obtiene una tasa de falsa aceptación distinta de cero. En cambio, si para las mismas configuraciones se utiliza el modo de verificación híbrida con el umbral de comodidad, se obtienen tasas de error nulas, **consiguiendo nivelar el rendimiento** de sus respectivos sistemas con los de las configuraciones resaltadas en color rojo.

En cuanto a las configuraciones resaltadas en amarillo, éstas representan aquellas configuraciones para las cuales no se pudieron obtener tasas de error nulas con ninguno de los dos modos de verificación. No obstante, utilizando el modo de verificación híbrida en lugar del tradicional, se consigue un **intercambio de error**. Para estas configuraciones,

cuando se establece el umbral de comodidad, se obtienen altas tasas de falsa aceptación utilizando el modo de verificación tradicional. En cambio, utilizando el modo híbrido, se reduce significativamente la tasa de falsa aceptación a expensas de aumentar la tasa de falso rechazo. En conclusión, el efecto que produce utilizar el modo de verificación híbrido sobre estas configuraciones es un **aumento de seguridad** en el sistema de reconocimiento.

Una vez obtenidos los resultados de la Tabla 10.3, surgieron una serie de preguntas:

Es importante comentar que los umbrales fijados en la Tabla 10.3 aseguran una tasa nula de falso rechazo **únicamente** para los sets de evaluación considerados, pero ¿y si fueran sistemas preparados para ser implementados en una aplicación real y lo único que se supiera de ellos fueran sus curvas FAR y FRR en función del umbral para un set de validación y sus **estimaciones** a una distribución estadística conocida, tal y como lo muestra el programa de evaluación implementado? ¿No sería mejor fijar el umbral de comodidad basándose en las **estimaciones** que en los resultados obtenidos de forma empírica?

Nótese que las curvas de FAR y FRR en función del umbral para los datos obtenidos de forma empírica en los sistemas que consiguen la separación de dichas curvas, no lo consiguen para las estimaciones (Figura 10.6, Figura 10.8, Figura 10.20, Figura 10.25). En consecuencia, al fijar el umbral de comodidad basándose en las estimaciones, es posible que se obtengan tasas de falsa aceptación distintas de cero sobre los datos obtenidos de forma empírica utilizando el modo de verificación tradicional. Dicho esto, en la Tabla 10.4 se observan los resultados obtenidos fijando el umbral de comodidad basándose en las estimaciones.

	Configuración	Umbral	Tradicional		Híbrido	
			FRR (%)	FAR (%)	FRR (%)	FAR (%)
YALE	PCA-medias-80%-0	0,45	0	3,8095	0	0
	LDA-medias-1	4,50	0	0	0	0
	GW-(6,8)-2	0,70	0	92,3810	0	0
FEI	PCA-muestras-99%-0	1,05	0	45,2381	6,6667	0,4762
	LDA-medias-0	5,60	0	0,4762	0	0
	GW-(6,8)-2	0,77	0	50,4762	0	0
PERSONAL	PCA-medias-99%-0	0,95	0	78,0952	33,3333	2,3810
	LDA-medias-0	2,00	0	29,0476	0	0
	GW-(10,10)-2	0,66	0	84,7619	40	2,8571

Tabla 10.4. Comparación entre el modo de verificación tradicional y el híbrido. Las tasas fueron calculadas sobre los datos obtenidos de forma empírica, y se fijó el umbral de comodidad basándose en las estimaciones.

Es importante hacer énfasis en el hecho de que las tasas mostradas en la Tabla 10.4 fueron calculadas sobre los datos obtenidos de forma empírica. Lo único que cambia entre las mediciones de Tabla 10.3 y la Tabla 10.4 es el umbral escogido, siendo en la primera el

umbral de comodidad de los datos obtenidos de forma empírica, y en la segunda el umbral de comodidad de las estimaciones.

La configuración del sistema basado en LDA (umbral por medias y sin pre-procesado) fue resaltada con color rojo debido a que con dicha configuración se obtuvieron tasas de error nulas para ambos modos de verificación. Este resultado era de esperar puesto que en su respectiva gráfica de FAR y FRR en función del umbral (Figura 10.8) se puede observar que para el umbral de comodidad de las estimaciones, no existen tasas de error para los datos obtenidos de forma empírica.

Las configuraciones resaltadas en color verde claro representan aquellas para las cuales se obtuvo una tasa de falsa aceptación distinta de cero usando el modo de verificación tradicional, y tasas nulas usando el modo de verificación híbrido.

Las configuraciones resaltadas en color verde oscuro representan aquellas para las cuales las tasas de falsa aceptación superaron el 50% utilizando el modo de verificación tradicional y, utilizando el modo de verificación híbrido, se consiguieron tasas nulas.

Para terminar, y al igual que en la Tabla 10.3, las configuraciones resaltadas en color amarillo representan aquellas para las cuales la utilización del modo de verificación híbrido supone un intercambio de error (aumento de seguridad).

Cabe destacar que los resultados obtenidos hasta el momento tenían el objetivo de confirmar que el modo de verificación híbrido reduce las tasas de falsa aceptación (aumenta la seguridad del sistema de reconocimiento). A continuación, se comparan el modo de verificación híbrido y el modo de verificación tradicional trabajando con un umbral que asegure tasas de falsa aceptación nulas, al cual se le llamará **umbral de seguridad**. El nombre escogido se debe a que cuando un sistema trabaja con un umbral de este tipo, se prefieren tasas de falso rechazo distintas de cero a cambio de obtener tasas de falsa aceptación muy bajas.

En la Tabla 10.5 se observan las tasas de falso rechazo y falsa aceptación calculadas sobre los datos obtenidos de forma empírica, y fijando el umbral de seguridad para las estimaciones. Nótese que las tasas de error utilizando el modo híbrido son las mismas que para el modo tradicional en este caso.

	Configuración	Umbral	Tradicional		Híbrido	
			FRR (%)	FAR (%)	FRR (%)	FAR (%)
YALE	PCA-medias-80%-0	0,26	60,0000	0	60,0000	0
	LDA-medias-1	4,00	0	0	0	0
	GW-(6,8)-2	0,85	20,0000	0	20,0000	0
FEI	PCA-muestras-99%-0	0,50	33,3333	0	33,3300	0
	LDA-medias-0	5,00	0	0	0	0
	GW-(6,8)-2	0,85	6,6667	0	6,6667	0
PERSONAL	PCA-medias-99%-0	0,40	100,000	0	100,000	0
	LDA-medias-0	1,20	86,6667	0	86,6667	0
	GW-(10,10)-2	0,84	100,000	0	100,000	0
			0	0	0	0

Tabla 10.5. Comparación entre el modo de verificación tradicional y el híbrido. Las tasas fueron calculadas sobre los datos obtenidos de forma empírica, y se fijó el umbral de seguridad basándose en las estimaciones.

Las configuraciones sin resaltar son aquellas donde se mantienen tasas de error nulas.

Las configuraciones resaltadas en color rojo son aquellas con las que se consiguieron tasas de falso rechazo distintas de cero. Sin embargo, al ser comparadas con los resultados obtenidos utilizando el modo híbrido y umbral de comodidad basado en las estimaciones (Tabla 10.4), resultan poco preferibles.

Las configuraciones resaltadas en color amarillo representan las configuraciones que quedan obsoletas fijando el umbral de seguridad (una tasa de falso rechazo del 100% indica que el sistema no admite a nadie).

Para terminar la descripción de la Tabla 10.4, la única configuración resaltada en color verde es aquella para la cual se consiguió una tasa de falsa aceptación nula (y menor que para el modo híbrido con umbral de comodidad de la Tabla 10.4). No obstante, la tasa de falso rechazo obtenida con esta configuración es muchísimo mayor que la obtenida para el modo híbrido con umbral de seguridad basado en las estimaciones (Tabla 10.4).

El inconveniente que presenta la Tabla 10.5 es justamente el motivo por el cual se optó por hacer estimaciones: falta de información relativa a los individuos legítimos. Como consecuencia, se decidió hacer una última prueba:

En la Tabla 10.6 se muestran las tasas de falsa aceptación y falso rechazo calculadas sobre las estimaciones, y utilizando el método de verificación tradicional. El umbral utilizado fue aquel con el que se obtendría una tasa de falsa aceptación igual a la del modo híbrido de la Tabla 10.4. Para este último procedimiento no fue posible calcular las tasas de error utilizando el modo híbrido, ya que éstas no pueden ser calculadas a partir de las estimaciones (se recuerda que el modo de verificación híbrido utiliza la lista de coincidencias del modo identificación).

	Configuración	Umbral	FRR (%)	FAR (%)
YALE	PCA-medias-80%-0	0,2601	58,97	0
	LDA-medias-1	4,052	0,1801	0
	GW-(6,8)-2	0,699	91,05	0
FEI	PCA-muestras-99%-0	0,6126	26,65	4,7619
	LDA-medias-0	4,712	2,941	0
	GW-(6,8)-2	0,7655	55,17	0
PERSONAL	PCA-medias-99%-0	0,5115	76,8	2,381
	LDA-medias-0	1,155	86,37	0
	GW-(10,10)-2	0,698	55,3	2,8571

Tabla 10.6. FAR y FRR obtenidas con el modo de verificación tradicional sobre las estimaciones. El umbral escogido fue aquel para el cual las FAR resultaban iguales a las del modo de verificación híbrido de la Tabla 10.4.

Si se comparan la Tabla 10.4, y la Tabla 10.6, resulta preferible utilizar el modo de verificación híbrido frente al modo de verificación tradicional. Nótese que para las mismas tasas de falsa aceptación, las tasas de falso rechazo son mayores para el modo tradicional que para el modo híbrido. Sin embargo, estos resultados **no son concluyentes** debido a que no existieron datos suficientes para poder evaluar el modo de verificación híbrido propuesto.

Una solución a este problema podría ser utilizar una base de datos lo suficientemente amplia como para poder evaluar el modo de verificación híbrido fijando el umbral de seguridad basado en las estimaciones.

Dicho esto, y tomando en cuenta los resultados obtenidos, se plantea la siguiente hipótesis:

Para el supuesto de que en una aplicación real las muestras mantengan una distribución estadística parecida a la de las muestras del entrenamiento, si se fija el umbral de comodidad basado en las estimaciones, es posible obtener tasas de falsa aceptación notablemente bajas sin necesidad de que las tasas de falso rechazo sean excesivamente altas utilizando el modo de verificación híbrido en lugar del modo de verificación tradicional.

Una última consideración a tomarse en cuenta cuando desee utilizarse el modo de verificación híbrido es que el sistema de clasificación utilizado debe tener un alto rendimiento en modo identificación (se recuerda que el modo híbrido clasifica a un individuo como legítimo únicamente si éste se encuentra de primero en la lista de coincidencias).

10.5. Evaluación de sistema de reconocimiento facial basado en EGM

Por el mismo motivo de que el sistema de reconocimiento basado en EGM se implementó únicamente en modo identificación (sobre todo por el hecho de que consume mucho tiempo cada clasificación, ver capítulo 9.9), no fue posible generar las curvas de

falsa aceptación y falso rechazo. A cambio, se utilizó un criterio de evaluación basado en la posición que ocupaba la identidad correcta de la imagen de evaluación de cada individuo en la lista de coincidencias del modo identificación.

La base de datos utilizada fue la de *Yale University* (15 individuos) [21]. Se generó el modelo de cada individuo a partir de una imagen donde sonreían, y la imagen de evaluación utilizada fue una donde el individuo hacía una mueca. Un ejemplo de las imágenes utilizadas se puede observar en la Figura 10.39.



Figura 10.39. Ejemplo de las imágenes utilizadas para la evaluación del sistema basado en EGM. La de la izquierda fue utilizada para generar el modelo, y la derecha fue utilizada como imagen de validación.

La evaluación se llevó a cabo para 3 grafos diferentes. Estos grafos pueden observarse en la Figura 10.40.

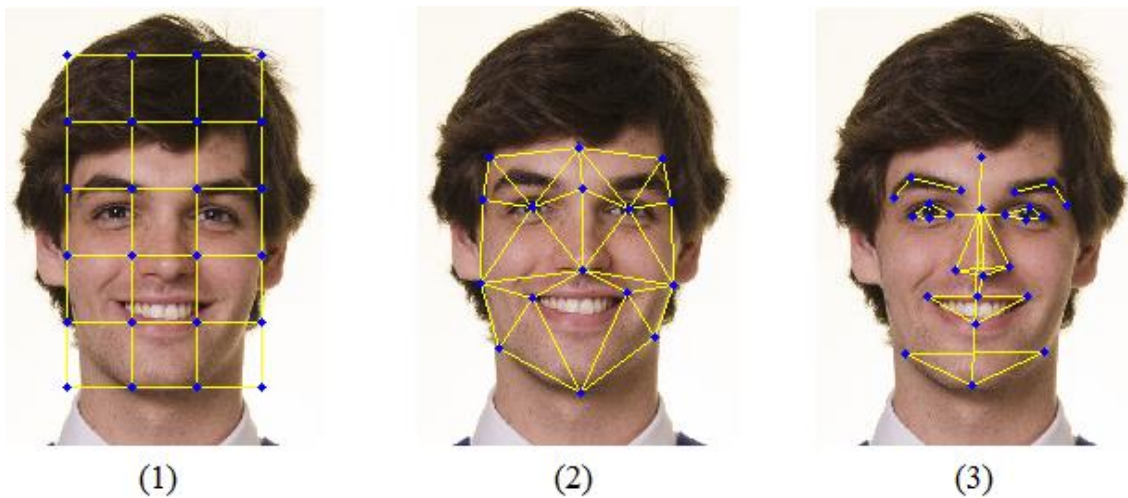


Figura 10.40. Grafos utilizados.

Los parámetros con los que se configuró el EGM, fueron los parámetros que se establecieron por defecto en el programa (Tabla 9.6).

El criterio de evaluación utilizado consistió en poner a prueba el sistema para la imagen de validación de cada individuo, obteniendo 5 posibles resultados:

- Resultado tipo A: La identidad correcta de la imagen de validación ocupó la primera posición de la lista de coincidencias
- Resultado tipo B: La identidad correcta de la imagen de validación ocupó la segunda posición de la lista de coincidencias

- Resultado tipo C: La identidad correcta de la imagen de validación ocupó la tercera posición de la lista de coincidencias.
- Resultado tipo D: La identidad correcta de la imagen de validación ocupó la cuarta posición de la lista de coincidencias.
- Resultado tipo E: La identidad correcta de la imagen de validación no ocupó ninguna de las 4 primeras posiciones de la lista de coincidencias

La Figura 10.7 muestra los resultados obtenidos para cada uno de los grafos.

	Resultados tipo A	Resultados tipo B	Resultados tipo C	Resultados tipo D	Resultados tipo E
Grafo 1	11	1	1	0	2
Grafo 2	9	4	1	0	1
Grafo 3	11	2	1	0	1

Tabla 10.7. Resultados obtenidos de la evaluación del sistema de reconocimiento basado en EGM.

Obsérvese que para el primer grafo el número de resultados tipo A se obtuvo para 11 de los 15 individuos, mientras que el número de identidades correctas que resultaron por debajo de la cuarta posición fue de 2 individuos (13,33%).

Para el segundo grafo se consiguieron menos resultados de tipo A que para el primero. Sin embargo, el número de identidades correctas que resultaron por debajo de la cuarta posición fue menor (1 sola, el 6,67%).

Por último, para el tercer grafo se obtuvieron los mejores resultados, ya que el número de identidades correctas que ocuparon la primera posición fue el mismo que para el primer grafo (11 casos), y el número de identidades correctas por debajo de la cuarta posición fue igual que para el segundo grafo (1 caso).

Tomando en cuenta los resultados obtenidos, el mérito que tiene la implementación del sistema de reconocimiento basado en EGM frente a los otros sistemas es que éste utilizó una única imagen de entrenamiento (un modelo por individuo), mientras que para la evaluación de los otros métodos se usó entre 5 y 10 imágenes de entrenamiento. Dicho esto, si se hubiese implementado el sistema de reconocimiento basado en EGM de tal manera que se pudiese crear más de un modelo por individuo, posiblemente se obtuvieran resultados comparables con los otros sistemas. No obstante, es importante considerar que los resultados de la Tabla 10.7 se obtuvieron después de unas varias horas, por lo tanto si por cada individuo se considerase más de un modelo, el tiempo de ejecución aumentaría considerablemente.

11. Conclusiones

En base a los objetivos planteados y los resultados obtenidos se extrajeron las siguientes conclusiones:

- Para las bases de datos utilizadas, y para las condiciones comentadas en cada una de ellas, el sistema de reconocimiento implementado con el que se obtuvieron los mejores resultados en general fue el basado en análisis lineal discriminante (hasta unas tasas de error nulas para el set de evaluación utilizado en este proyecto).
- Como era de esperar, los resultados obtenidos para una base de datos creada a partir de imágenes con condiciones de adquisición controladas fueron más satisfactorios que para una base de datos creada a partir de imágenes con condiciones de adquisición no controladas. Sin embargo, los resultados obtenidos para la base de datos creada a partir de imágenes no controladas pueden ser considerados como aceptables (tasas de error por debajo del 5%).
- A pesar de haber creado un único modelo por cada individuo en el sistema implementado de reconocimiento facial basado en EGM, se obtuvieron resultados que sugieren buenas expectativas para una implementación en la que se aceptasen más modelos por cada individuo. No obstante, estas implementaciones tendrían la desventaja de una mayor demora en la clasificación.
- Utilizando el modo de verificación híbrido se obtuvieron mejoras significativas en cuanto al modo de verificación tradicional fijando el umbral de comodidad. No obstante, esto no implica que el modo de verificación híbrido propuesto sea mejor que el modo de verificación tradicional en todos los casos, ya que los resultados obtenidos no son concluyentes en cuanto al umbral de seguridad.
- Es posible mejorar el rendimiento de un sistema de reconocimiento facial utilizando el pre-procesado adecuado. Por ejemplo, para las configuraciones medidas en este proyecto se consiguió reducir el EER desde un 1% hasta un 0,07% aplicando solamente una ecualización de histograma.
- Se han implementado 4 sistemas de reconocimiento facial basados en esquemas competitivos, pero que requieren cierto control de adquisición (posición, iluminación, etc.). Habría que hacer un estudio detallado sobre la influencia de este control en los resultados. Sin embargo, este no era el objetivo de este proyecto de fin de grado.

12. Principales contribuciones

En este capítulo se enumeran las contribuciones principales de este proyecto:

- Utilización del porcentaje de varianza total representado por cada eigenface resultante del análisis por componentes principales. **De esta forma se consigue la posibilidad de descartar eigenfaces de manera objetiva.** Además, se consigue controlar el número de *eigenfaces* utilizando un rango de valores más cómodo (porcentajes de varianza total mínima) en lugar de rango de valores variable (el número de *eigenfaces* máximo que se puede utilizar en una clasificación es variable en función de la base de datos).
- Propuesta de un modo de verificación híbrido que utiliza los resultados de la lista de coincidencias del modo de identificación para llevar a cabo la verificación de identidad un individuo.
- La implementación cuenta con un programa de evaluación de los sistemas desarrollados. Adicionalmente, dicho programa de evaluación está dotado con la característica de permitir al usuario elegir la estimación de la distribución original de los datos a una distribución estadística conocida según su propio criterio.
- El programa implementado en MATLAB ofrece la posibilidad de crear una base de datos de rostros etiquetados de manera muy intuitiva.
- La implementación de la herramienta interactiva de creación de grafos en conjunto con la herramienta de creación de modelos a partir del grafo creado son sencillas de manejar, y pueden incluso ser separadas del programa para otros fines ajenos al reconocimiento facial.
- El programa permite al usuario elegir un porcentaje de corte adicional durante el proceso de segmentación de rostros con el fin de ahorrar tiempo y conseguir segmentaciones más uniformes entre distintas imágenes.
- En la implementación del sistema de reconocimiento facial basado en *wavelets* de Gabor se propuso un método para que el tiempo de espera del entrenamiento fuese constante independientemente del número de imágenes de entrenamiento.
- La implementación del sistema de reconocimiento basado en emparejamiento de grafos elásticos puede ser válida para el reconocimiento de otros tipos de objetos [36] [31].
- Se propone una solución al problema surgido con respecto a la realimentación al usuario del estado de la clasificación del sistema de reconocimiento facial basado en emparejamiento de grafos elásticos (EGM) mostrándole el grafo deformándose en tiempo real.
- Gracias a la estructura de diseño ordenada del programa, es posible efectuar ampliaciones del mismo sin la necesidad de cambiar apenas las funciones de la versión actual.

- La implementación hecha en MATLAB está programada de tal manera que sea sencillo separar las funciones y utilizar el código fuente para futuras investigaciones.
- Las explicaciones expuestas en el marco teórico son la interpretación personal del autor de este proyecto, con lo que la gran mayoría de las figuras (muchas de ellas generadas en MATLAB) representan un aporte desde el punto de vista teórico.

13. Líneas futuras de investigación

- Hacer un estudio más profundo sobre el modo de verificación híbrido propuesto con el fin de dar una respuesta a la hipótesis planteada al final del capítulo 10.4 (su utilización podría mejorar el rendimiento de un sistema trabajando con el umbral de seguridad y el modo de verificación tradicional).
- Ampliar la implementación dando al usuario la posibilidad de elegir las frecuencias y rotaciones de las *wavelets* utilizadas en la implementación del sistema basado en *wavelets* de Gabor.
- Hacer un estudio sobre los pre-procesados de imagen que pueden ser beneficiosos para el reconocimiento facial, como por ejemplo mejoras de ruido sobre las imágenes adquiridas en ambientes no controlados.
- Ampliar el funcionamiento del programa implementado en MATLAB añadiendo otras técnicas de reconocimiento facial, como por ejemplo *support vector machines*, *redes neuronales*, o incluso modelos en 3D.
- Utilizar una estructura de diseño parecida a la de esta implementación para desarrollar otros sistemas de reconocimiento biométrico, como reconocimiento de voz, o huella dactilar.

Referencias

- [1] X. Lu, *Image Analysis for Face Recognition*, 2003.
- [2] R. G. Hernández, *Estudio de técnicas de reconocimiento facial*, Barcelona, España, 2010.
- [3] B. Miller, «Vital signs of identity,» *IEEE Spectrum*, vol. 31, nº 20-30, pp. 22-30, 1994.
- [4] S. N. Srihari y H. Srinivasan, «Comparison of ROC and Likelihood Decision Methods in Automatic Fingerprint Verification,» *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, nº 3, pp. 535-553, 2008.
- [5] A. Martin, G. Doddington, T. Kamm, M. Ordowski y M. Przybocki, «The DET Curve in Assessment of Detection Task,» de *Proceedings of Eurospeech*, Rhodes, Greece, 1997.
- [6] D. Simón Zorita, Reconocimiento automático mediante patrones biométricos de huella dactilar, Madrid, España: Tesis doctoral de la Universidad Politécnica de Madrid, 2003.
- [7] Y.-J. Chen y Y.-C. Lin, «Simple Face-detection Algorithm Based on Minimum Facial Features,» de *The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Taipei, Taiwan, 2007.
- [8] S. K. Singh, D. S. Chauhan, M. Vatsa y R. Singh, «A Robust Skin Color Based Face Detection Algorithm,» *Tamkang Journal of Science and Engineering*, vol. 6, nº 4, pp. 227-234, 2003.
- [9] M. Turk y A. Pentland, «Eigenfaces for Recognition,» *Journal of Cognitive Neuroscience*, vol. 3, nº 1, pp. 71-86, 1991.

- [10] E. Saber y M. Tekalp, «Frontal-View Face Detection and Facial Feature Extraction using Color, Shape and Symmetry Based Cost Functions,» *Pattern Recognition Letters*, vol. 19, n° 8, pp. 669-680, 1998.
- [11] S. A. Sirohey, *Human Face Segmentation and Identification*, College Park, United States, 1993.
- [12] A. L. Yuille, «Deformable Templates for Face Recognition,» *Journal of Cognitive Neuroscience*, vol. 3, n° 1, pp. 59-70, 1991.
- [13] H. A. Rowley, S. Baluja y T. Kanade, «Neural Network-Based Face Detection,» *IEEE TPAMI*, vol. 20, n° 1, pp. 23-38, 1998.
- [14] P. Viola y M. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features,» de *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.
- [15] M. Kearns, *Thoughts on Hypothesis Boosting*, 1998.
- [16] Y. Freund y R. E. Schapire, «A decision-theoretic generalization of on-line learning and an application to boosting,» *Journal of Computer and System Sciences*, vol. 55, n° 1, pp. 119-139, 1997.
- [17] R. E. Schapire y Y. Singer, «Improved Boosting Algorithms Using Confidence-rated Predictions,» *Machine Learning*, vol. 37, n° 3, pp. 297-336, 1999.
- [18] R. Lienhart y J. Maydt, «An Extended Set of Haar-like Features for Rapid Object Detection,» de *International Conference on Image Processing*, New York, United States, 2002.
- [19] O. H. Jensen, *Implementing the Viola-Jones Face Detection Algorithm*, Kongens Lyngby, 2008.
- [20] D. C. Lay, *Álgebra Lineal y sus aplicaciones*, Pearson, 2006.

- [21] Y. University. [En línea]. Available: <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>. [Último acceso: diciembre 2013].
- [22] P. N. Belhumeur, J. P. Hespanha y D. J. Kriegman, «Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,» *IEEE TPAMI*, vol. 19, nº 7, pp. 711-720, 1997.
- [23] A. Justel, *Técnicas de análisis discriminante*, Madrid, España.
- [24] R. O. Duda, P. E. Hart y D. G. Stork, *Pattern Classification*, John Wiley and Sons, 2001.
- [25] A. V. Oppenheim, A. S. Willsky y W. S. Hamid, *Signals and Systems*, Prentice-Hall, 1996.
- [26] L. Chun-Lin, *A Tutorial of the Wavelet Transform*, 2010.
- [27] R. Polikar, *The wavelet tutorial*.
- [28] C. Vales, *A Really Friendly Guide to Wavelets*, 1999.
- [29] J. R. Movellan, *Tutorial on Gabor Filters*, 2008.
- [30] L. Shen y L. Bai, «A review on Gabor wavelets for face recognition,» *Pattern Analysis and Applications*, vol. 9, nº 2-3, pp. 273-292, 2006.
- [31] L. Wiskott, J.-M. Fellous, N. Krüger y C. von der Malsburg, «Face Recognition by Elastic Bunch Graph Matching,» *IEEE TPAMI*, vol. 19, nº 7, pp. 775-779, 1997.
- [32] T. S. Lee, «Image Representation Using 2D Gabor Wavelets,» *TPAMI*, vol. 18, nº 10, pp. 959-971, 1996.
- [33] C. Aguerrebere y G. Capdehourat, *Reconocimiento de caras con características locales*, Montevideo, Uruguay, 2006.

- [34] D. S. Bolme, *Elastic Bunch Graph Matching*, Fort Collins, United States, 2003.
- [35] K. B. Vinay, *Face Recognition Using Gabor Wavelets*, Bangalore, Inddia, 2006.
- [36] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R. P. Würtz y W. Konen, «Distortion Invariant Object Recognition in the Dynamic Link Architecture,» *IEEE Transactions on Computers*, vol. 42, nº 3, pp. 300-311, 1993.
- [37] Centro Universitário da FEI, «FEI Face Database,» [En línea]. Available: <http://fei.edu.br/~cet/facedatabase.html>. [Último acceso: diciembre 2013].
- [38] MathWorks, MATLAB® Creating Graphical User Interfaces, 2013.

ÍNDICE DE FIGURAS

<i>Figura 2.1. Diagrama de bloques genérico de un sistema biométrico.</i>	5
<i>Figura 2.2. Funciones densidad de probabilidad de individuos legítimos e impostores.</i>	7
<i>Figura 2.3. Tasas de falsa aceptación y falso rechazo en función del umbral de decisión.</i>	8
<i>Figura 2.4. Curva ROC.</i>	8
<i>Figura 2.5. Curva DET.</i>	9
<i>Figura 3.1. Diagrama de bloques general de un sistema de reconocimiento facial.</i>	11
<i>Figura 4.1. Algoritmo AdaBoost.</i>	16
<i>Figura 4.2. Demostración gráfica del algoritmo AdaBoost.</i>	18
<i>Figura 4.3. Tipos de características de Haar usadas por Viola-Jones.</i>	19
<i>Figura 4.4. Cálculo de tres de las característica de una matriz.</i>	19
<i>Figura 4.5. Zonas de una imagen cubiertas por los valores de su imagen integral en las posiciones a, b, c y d.</i>	20
<i>Figura 4.6. AdaBoost modificado de Viola-Jones.</i>	22
<i>Figura 4.7. Los dos mejores clasificadores simples que obtuvieron Viola y Jones en su implementación.</i>	24
<i>Figura 4.8. Clasificadores en cascada. Las porciones que superen la última etapa serán aquellas clasificadas como caras por el detector.</i>	25
<i>Figura 5.1. Procedimiento de cálculo de las componentes principales. Las gráficas fueron obtenidas generando una distribución aleatoria en MATLAB y calculando sus componentes principales.</i>	31
<i>Figura 5.2. Proyección de los datos de la Figura 5.1 sobre cada una de las componentes principales.</i>	32
<i>Figura 5.3. Formación de un vector imagen.</i>	33
<i>Figura 5.4. Set de entranamiento formado por algunas de las imágenes de la base de datos de Yale University [21].</i>	36
<i>Figura 5.5. Imagen media del set de entrenamiento de la Figura 5.4. Esta figura fue obtenida utilizando MATLAB.</i>	36
<i>Figura 5.6. Veinte primeras eigenfaces del set de entrenamiento de la Figura 5.4. Esta figura fue obtenida utilizando MATLAB.</i>	37
<i>Figura 6.1 Diagrama de dispersión de dos clases cuyas muestras fueron generadas aleatoriamente en MATLAB.</i>	39
<i>Figura 6.2. Análisis por componentes principales y posterior reducción de la dimensión de los datos de la Figura 6.1.</i>	39
<i>Figura 6.3. Demostración de la maximización de la distancia entre clases para las muestras de la Figura 6.1.</i>	40
<i>Figura 6.4. Minimización de la varianza dentro de las clases para las muestras de la Figura 6.1.</i>	41
<i>Figura 6.5. Casos de reducción de dimensión según el eje que maximiza la distancia entre clases y según el eje que minimiza la varianza dentro de clases. Al igual que las figuras anteriores, tanto la distribución aleatoria de las muestras, como sus proyecciones sobre los ejes indicados fueron generadas en MATLAB.</i>	42
<i>Figura 6.6. Ejemplo de una distribución de 3 clases donde se comprueba que el LDA garantiza mayor separabilidad entre clases que el PCA.</i>	44
<i>Figura 6.7. Fisherfaces del set de entrenamiento de la Figura 5.4.</i>	45
<i>Figura 7.1. Relación entre resolución frecuencial y temporal para (a) la transformada de Fourier, (b) para una STFT con tamaño de ventana medio, (c) para una STFT de banda estrecha (ventana ancha) y (d) para una STFT de banda ancha (ventana estrecha).</i>	48
<i>Figura 7.2. Relación entre resolución frecuencial y temporal para el análisis multirresolución.</i>	49
<i>Figura 7.3. Wavelet de Ricker. También conocida como Mexican hat wavelet (wavelet sombrero mexicano).</i>	50
<i>Figura 7.4. Wavelet de Ricker para diferentes escalas. La escala $s = 1$ es la wavelet madre. Todas las gráficas fueron generadas en MATLAB.</i>	50
<i>Figura 7.5. Gráficas de 4 exponenciales complejas bidimensionales con diferentes parámetros.</i>	52

Figura 7.6. Gráfica de dos envolventes gaussianas con distintos parámetros. Los parámetros indicados en la primera gaussiana son tomando en cuenta que el origen de coordenadas está en el centro de la imagen. Aunque no se indique, la segunda gaussiana está centrada en las coordenadas (0,0).	53
Figura 7.7. Filtros de Gabor con distintos parámetros. El centro de coordenadas está en el origen de la imagen.	54
Figura 7.8. A la izquierda se observan filtros de Gabor unidimensionales a distintas frecuencias con σ fijo, y a la derecha otros filtros con las mismas frecuencias que los de la izquierda, pero con σ dependiente de $F0$.	55
Figura 7.9. Wavelets de Gabor obtenidas con MATLAB para 5 frecuencias, 8 rotaciones. Área de 128x128 píxeles.	57
Figura 7.10. Representación tridimensional de una wavelet de Gabor obtenida con MATLAB.	57
Figura 7.11. Resultado de convolucionar 3 wavelets distintas con imágenes. Tanto las wavelets, como la magnitud de las convoluciones fueron obtenidas en MATLAB.	58
Figura 7.12. Características del individuo que se muestra en la fotografía, calculadas convolucionando la imagen con las wavelets de Gabor de la Figura 7.9. Estas gráficas, al igual que todas las anteriores, fueron obtenidas en MATLAB.	59
Figura 7.13. Ejemplo de una malla de 5x5 puntos sobre una imagen facial y las magnitudes del resultado de su convolución con una wavelet de Gabor. Las características a almacenar serán los valores de las matrices de características que coincidan con los puntos rojo.	60
Figura 8.1. Grafos utilizados por Lades et al. [36] (a la izquierda) y Bolme [34] (a la derecha) en sus trabajos.	63
Figura 8.2. Asociación de un grafo a una imagen.	64
Figura 8.3. Recolocación de un vértice.	66
Figura 9.1. Diagrama de bloques general de los módulos de reconocimiento implementados.	70
Figura 9.2. Esquema de navegación del programa de reconocimiento facial.	72
Figura 9.3. Estructura de diseño del programa.	73
Figura 9.4. Ejemplos de detección y segmentación para una misma imagen.	75
Figura 9.5. Tipos de umbral utilizados.	78
Figura 9.6. Problema que surge cuando se establecen varios umbrales originales (arriba) y solución estableciendo un umbral original único (abajo).	79
Figura 9.7. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en PCA. El bloque común es el mostrado en la Figura 9.1.	79
Figura 9.8. Ventana de propiedades del sistema de reconocimiento basado en PCA.	80
Figura 9.9. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en LDA. El bloque común es el mostrado en la Figura 9.1.	81
Figura 9.10. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en wavelets de Gabor. El bloque común es el mostrado en la Figura 9.1.	83
Figura 9.11. A la izquierda una malla de puntos superpuesta a una imagen, en el centro la parte real de una wavelet de Gabor, y a la derecha la misma wavelet desplazada al punto superior izquierdo de la malla. Para poder llevar a cabo la proyección, la wavelet trasladada debe ser del mismo tamaño que la imagen.	84
Figura 9.12. Diagrama de bloques de la implementación del sistema de reconocimiento facial basado en EGM. El bloque común es el mostrado en la Figura 9.1.	86
Figura 9.13. Herramienta de creación de grafos.	88
Figura 9.14. Ejemplos de un grafos predefinidos creados con la herramienta de creación de grafos programada.	88
Figura 9.15. Herramienta de creación de modelos.	89
Figura 9.16. Ventana de visualización para el sistema de reconocimiento facial basado en PCA. Las eigenfaces que en ella se muestran son las correspondientes a una base de datos personal del autor.	92
Figura 9.17. Ventana de clasificación mostrando la primera coincidencia de la lista de coincidencias.	93
Figura 9.18. Ejemplos donde el programa clasifica al individuo de la imagen de entrada como un desconocido.	94

<i>Figura 9.19. Ventana de clasificación en modo identificación para el sistema basado en EGM.</i>	95
<i>Figura 9.20. Ventana de clasificación del modo verificación.</i>	95
<i>Figura 9.21. Estructura de diseño del programa con los directorios y función principal del programa de evaluación incluidos.</i>	97
<i>Figura 9.22. Menú principal del programa de evaluación.</i>	98
<i>Figura 9.23. Ventana de elección de la estimación de densidad de probabilidad.</i>	99
<i>Figura 9.24. Curvas FAR y FRR en función del umbral resultantes de un sistema. El programa de evaluación representa las curvas obtenidas de forma empírica como líneas finas, y las obtenidas mediante estimaciones como líneas gruesas.</i>	100
<i>Figura 9.25. Múltiples curvas ROC mostradas en una sola gráfica por el programa de evaluación.</i>	101
<i>Figura 10.1. Tipo de corte utilizado para segmentar la imágenes de la base de datos de Yale University.</i>	105
<i>Figura 10.2. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral a una resolución del 100% (BD:YALE).</i>	106
<i>Figura 10.3. Curvas ROC y DET para distintas resoluciones de un umbral por muestras y con ecualización de histograma como pre-procesado (BD:YALE).</i>	107
<i>Figura 10.4. Curvas ROC y DET para las resoluciones mínimas: 75%, 80%, 85% y 100% (BD:YALE).</i>	107
<i>Figura 10.5. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral a una resolución mínima del 50% (BD:YALE).</i>	108
<i>Figura 10.6. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en PCA (BD:YALE).</i>	109
<i>Figura 10.7. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral para un sistema basado en LDA (BD:YALE).</i>	110
<i>Figura 10.8. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en LDA (BD:YALE).</i>	110
<i>Figura 10.9. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado: Ecualización de histograma) (BD:YALE).</i>	111
<i>Figura 10.10. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado: Normalización de varianza). (BD:YALE).</i>	112
<i>Figura 10.11. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado: Ecualización de histograma y normalización de varianza)(BD:YALE).</i>	112
<i>Figura 10.12. Curvas ROC y DET para todos los posibles pre-procesados de la malla de puntos de 6x8 y para el pre-procesado mixto de la malla de 10x10 (BD:YALE).</i>	113
<i>Figura 10.13. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en wavelets de Gabor (BD:YALE).</i>	113
<i>Figura 10.14. Comparación general de los sistemas basados en PCA, LDA, y wavelets de Gabor (BD:YALE).</i>	114
<i>Figura 10.15. Tipo de corte utilizado para segmentar la imágenes de la base de datos del Centro Universitário da FEI.</i>	115
<i>Figura 10.16. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral a una resolución del 100% (BD:FEI).</i>	116
<i>Figura 10.17. Curvas ROC y DET para distintas resoluciones de un umbral por muestras con ecualización de histograma de pre-procesado (BD:FEI).</i>	116
<i>Figura 10.18. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en PCA (BD:FEI).</i>	117
<i>Figura 10.19. Curvas ROC y DET para todas las combinaciones posibles de tipo de pre-procesado y tipo de umbral para un sistema basado en LDA. No es posible representar de manera visible las cuatro ROC en una sola gráfica (BD:FEI).</i>	118
<i>Figura 10.20. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en LDA (BD:FEI).</i>	119
<i>Figura 10.21. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Ecualización de histograma) (BD:FEI).</i>	119

<i>Figura 10.22. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Normalización de varianza) (BD:FEI).</i>	120
<i>Figura 10.23. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado mixto) (BD:FEI).</i>	120
<i>Figura 10.24. Curvas ROC y DET para los posibles pre-procesados de la malla de puntos de 6x8 (BD:FEI).</i>	121
<i>Figura 10.25. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en wavelets de Gabor (BD:FEI).</i>	121
<i>Figura 10.26. Comparación general de los sistemas basados en PCA, LDA, y wavelets de Gabor (BD:FEI).</i>	122
<i>Figura 10.27. Ejemplo de una de las imágenes de la base de datos personal no-controlada.</i>	122
<i>Figura 10.28. Curvas ROC y DET de todas las configuraciones posibles para una resolución máxima (BD:PERSONAL).</i>	123
<i>Figura 10.29. Curvas ROC y DET para distintas resoluciones de una configuración de umbral por muestras y ecualización de histograma como pre-procesado (BD:PERSONAL).</i>	123
<i>Figura 10.30. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en PCA (BD:PERSONAL).</i>	124
<i>Figura 10.31. Curvas ROC y DET para todas las configuraciones posibles del sistema basado en LDA (BD:PERSONAL).</i>	124
<i>Figura 10.32. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en LDA (BD:PERSONAL).</i>	125
<i>Figura 10.33. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Ecualización de histograma) (BD:PERSONAL).</i>	125
<i>Figura 10.34. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (Normalización de varianza) (BD:PERSONAL).</i>	126
<i>Figura 10.35. Curvas ROC y DET para distintos tamaños de malla de puntos para wavelets de Gabor (pre-procesado mixto) (BD:PERSONAL).</i>	126
<i>Figura 10.36. Curvas ROC y DET de todos los pre-procesados posibles para la malla de puntos de 10x10 (BD: PERSONAL).</i>	127
<i>Figura 10.37. Densidades de probabilidad de individuos legítimos e impostores y curvas FAR y FRR en función del umbral para la configuración representante del sistema basado en wavelets de Gabor (BD:PERSONAL).</i>	127
<i>Figura 10.38. Comparación general de los sistemas basados en PCA, LDA, y wavelets de Gabor (BD:PERSONAL).</i>	128
<i>Figura 10.39. Ejemplo de las imágenes utilizadas para la evaluación del sistema basado en EGM. La de la izquierda fue utilizada para generar el modelo, y la derecha fue utilizada como imagen de validación....</i>	134
<i>Figura 10.40. Grafos utilizados.</i>	134

ÍNDICE DE TABLAS

<i>Tabla 9.1. Propiedades generales por defecto.</i>	76
<i>Tabla 9.2. Valores de cada columna de la matriz de actualización en función de la operación realizada...</i>	77
<i>Tabla 9.3. Propiedades por defecto del sistema de reconocimiento facial basado en PCA.</i>	80
<i>Tabla 9.4. Propiedades por defecto del sistema de reconocimiento facial basado en LDA.</i>	82
<i>Tabla 9.5. Propiedades por defecto del sistema de reconocimiento facial basado en wavelets de Gabor. ..</i>	83
<i>Tabla 9.6. Propiedades por defecto del sistema de reconocimiento facial basado en EGM.</i>	87
<i>Tabla 9.7. Nomenclatura del programa de evaluación.</i>	98
<i>Tabla 10.1. Porcentajes de corte adicional utilizados para la segmentación de rostros de la base de datos de Yale University</i>	104
<i>Tabla 10.2. Porcentajes de corte adicional utilizados para la segmentación de rostros de la base de datos del Centro Universitário da FEI.</i>	114
<i>Tabla 10.3. Comparación entre el modo de verificación tradicional y el híbrido. Las tasas fueron calculadas sobre los datos obtenidos de forma empírica, y se fijó el umbral de comodidad basándose en los datos obtenidos de forma empírica.</i>	129
<i>Tabla 10.4. Comparación entre el modo de verificación tradicional y el híbrido. Las tasas fueron calculadas sobre los datos obtenidos de forma empírica, y se fijó el umbral de comodidad basándose en las estimaciones.</i>	130
<i>Tabla 10.5. Comparación entre el modo de verificación tradicional y el híbrido. Las tasas fueron calculadas sobre los datos obtenidos de forma empírica, y se fijó el umbral de seguridad basándose en las estimaciones.</i>	132
<i>Tabla 10.6. FAR y FRR obtenidas con el modo de verificación tradicional sobre las estimaciones. El umbral escogido fue aquel para el cual las FAR resultaban iguales a las del modo de verificación híbrido de la Tabla 10.4.</i>	133
<i>Tabla 10.7. Resultados obtenidos de la evaluación del sistema de reconocimiento basado en EGM.</i>	135

ANEXO

MANUAL DE USUARIO

La implementación está formada por dos programas. El primero de ellos, el programa de reconocimiento, está compuesto por los siguientes sistemas de reconocimiento facial: Análisis por componentes principales (PCA, por sus siglas en inglés), análisis lineal discriminante (LDA, por sus siglas en inglés), *wavelets* de Gabor, y emparejamiento de grafos elásticos (EGM, por sus siglas en inglés). El segundo es el programa de evaluación, y se accede a él de forma independiente al programa de reconocimiento.

En consecuencia, este manual de usuario se encuentra dividido en tres partes: **Manual del programa de reconocimiento, Guía de utilización del programa de evaluación y Tablas de propiedades por defecto.**

En vista de que la implementación está programada en MATLAB, es necesario tener dicho software instalado. Es imprescindible disponer del *Image Processing Toolbox*. Adicionalmente, si se deseara utilizar la detección automática de rostros debe disponerse también del *Computer Vision System Toolbox*, de lo contrario se deshabilitarán todas las opciones de detección automática.

Algunas de las imágenes que aparecen en este manual de usuario forman parte de la base de datos de *Yale University* [1].

Índice del manual de usuario

Manual del programa de reconocimiento	5
Menú principal	5
Ventana de propiedades generales	6
Menú de creación y edición de la base de datos	7
Ventana de actualización y/o inscripción de individuos a la base de datos	8
Ventana de edición de individuos	9
Ventana de eliminación de individuos	10
Ventana de confirmación de identidad	11
Ventana de captura de imagen a través de cámara web	12
Herramienta de detección y segmentación de individuos (I)	13
Herramienta de detección y segmentación de individuos (II)	14
Herramienta de etiquetado de imágenes	15
Menú del sistema de reconocimiento facial basado en PCA	16
Ventana de propiedades del sistema basado en PCA	16
Ventana de visualización del sistema basado en PCA	17
Menú del sistema de reconocimiento facial basado en LDA	18
Ventana de propiedades del sistema basado en LDA	18
Ventana de visualización del sistema basado en LDA	19
Menú del sistema de reconocimiento facial basado en <i>wavelets</i> de Gabor	20
Ventana de propiedades del sistema basado en <i>wavelets</i> de Gabor	20
Ventana de visualización del sistema basado en <i>wavelets</i> de Gabor	21
Ventana de clasificación en modo verificación	22
Ventana de clasificación en modo identificación (I)	23
Ventana de clasificación en modo identificación (II)	24
Ventana de identificación fallida	25
Menú del sistema de reconocimiento facial basado en EGM	26

Ventana de propiedades del sistema basado en EGM	26
Ventana de visualización del sistema basado en EGM.....	27
Gestor de modelos.....	28
Herramienta de creación de grafos (I)	29
Herramienta de creación de grafos (II)	30
Herramienta de creación de grafos (III).....	31
Herramienta de creación de modelos a partir del grafo generado (I)	32
Herramienta de creación de modelos a partir del grafo generado (II)	33
Guía de utilización del programa de evaluación.....	35
Tabla de nomenclatura	35
Ventana principal del programa de evaluación.....	36
Ventana de creación del set de validación	37
Ventanas de selección de propiedades de evaluación y de cálculo de tasas.....	38
Visualización de las evaluaciones realizadas.....	39
Tablas de propiedades por defecto.....	41
Propiedades generales por defecto	41
Referencia citada en este manual de usuario	42

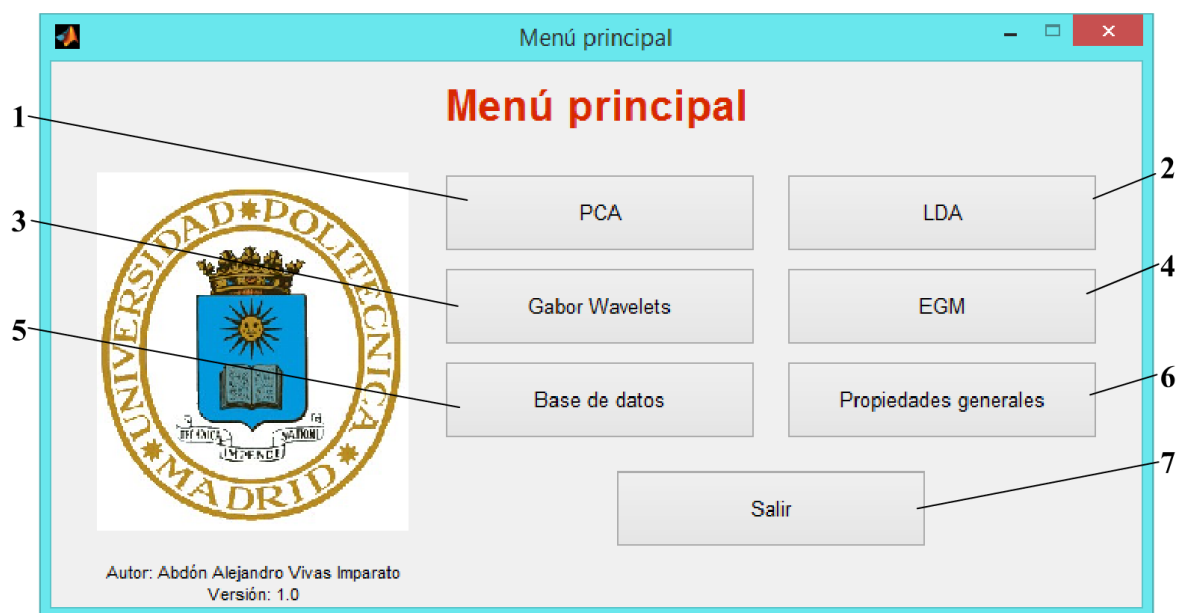
Manual del programa de reconocimiento

Para iniciar el programa de reconocimiento, debe seleccionarse el directorio raíz del mismo como el directorio actual y ejecutar la función principal (*main*) ya sea abriendo su código fuente y ejecutándola desde la barra de herramientas, o simplemente escribiendo *main* en la ventana de comandos. Una vez hecho esto, se abrirá la ventana principal del programa.

Si es la primera vez que se abre el programa, se recomienda cambiar primero las propiedades de las imágenes a almacenar en la base de datos en la ventana de **propiedades generales**, ya que éstas solo podrán ser modificadas antes de añadir la primera imagen a la base de datos. Posteriormente, debe crearse una base de datos, o de lo contrario se mantendrán deshabilitados todos los sistemas de reconocimiento facial.

Menú principal

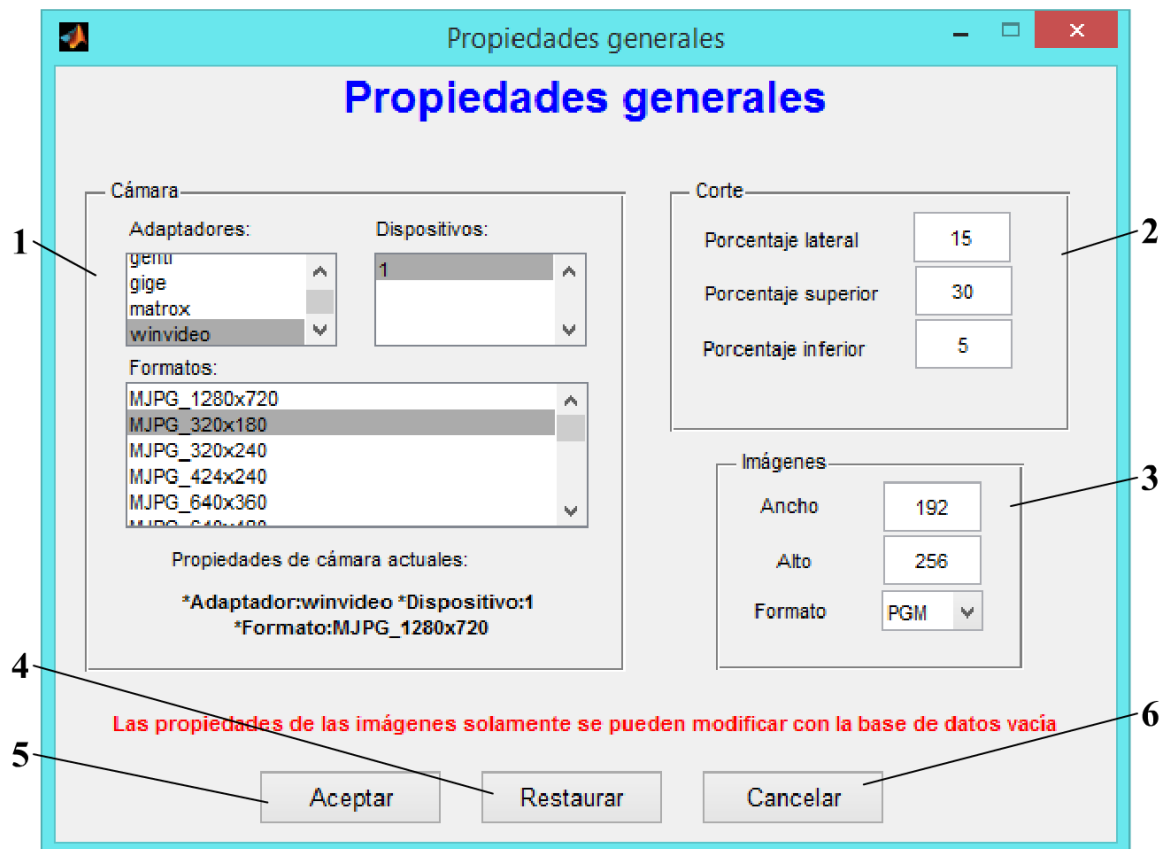
A través de él se accede a todas las funcionalidades del programa de reconocimiento.



1. Accede al menú correspondiente al sistema de reconocimiento facial basado en análisis por componentes principales (PCA).
2. Accede al menú correspondiente al sistema de reconocimiento facial basado en análisis lineal discriminante (LDA).
3. Accede al menú correspondiente al sistema de reconocimiento facial basado en *wavelets* de Gabor.
4. Accede al menú correspondiente al sistema de reconocimiento facial basado en emparejamiento de grafos elásticos (EGM).
5. Accede al menú de creación y edición de la base de datos.
6. Abre la ventana de propiedades generales.
7. Cierra el programa de reconocimiento.

Ventana de propiedades generales

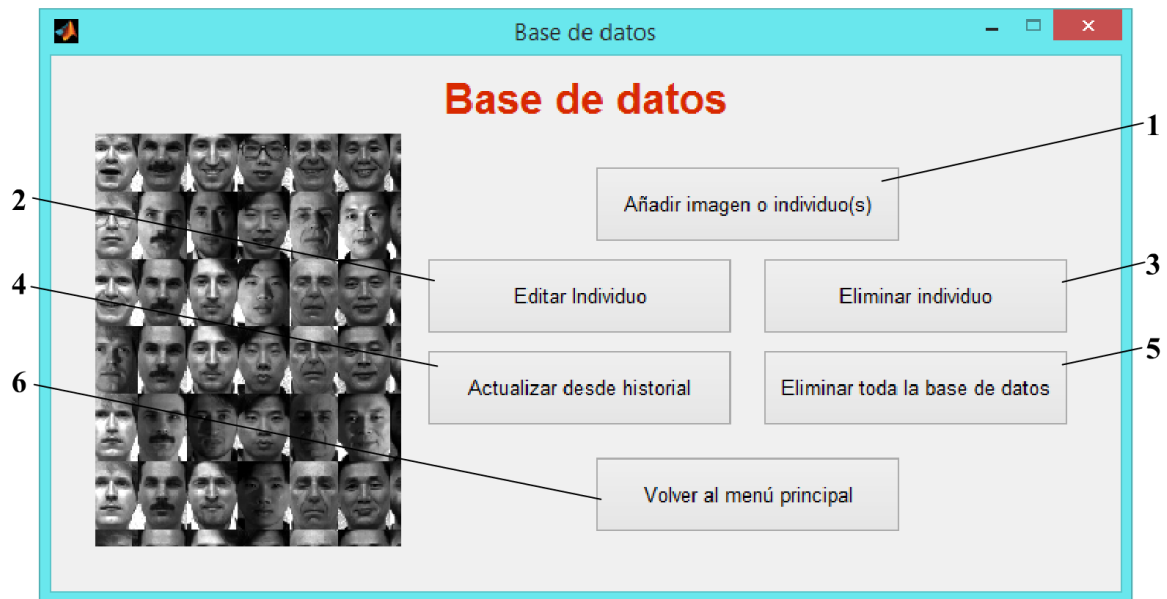
En ella se permite editar las propiedades generales del programa de reconocimiento.



1. Panel de propiedades de cámara: Permite seleccionar el dispositivo de captura y la resolución que se desee. En el caso de que no se detecte ningún dispositivo conectado, estas opciones aparecerán deshabilitadas.
2. Panel de porcentajes adicionales de corte a aplicar en la herramienta de segmentación de rostros. Se permiten porcentajes negativos.
3. Panel de propiedades de las imágenes a almacenar en la base de datos (Ancho, alto y formato). Solamente es posible elegir un formato entre los siguientes: PMG, JPG, PNG y GIF.
4. Restaura las propiedades por defecto.
5. Confirma los cambios realizados.
6. Cancela las modificaciones y cierra la ventana.

Menú de creación y edición de la base de datos

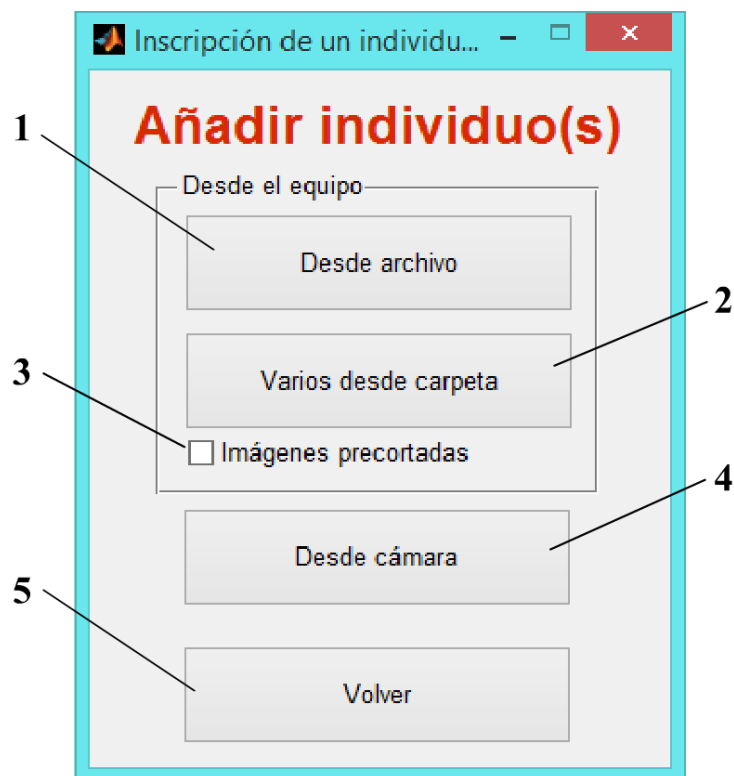
A través de él se puede acceder a las ventanas de creación y edición de la base de datos.



1. Accede a la ventana de actualización y/o inscripción de individuos a la base de datos.
2. Abre la ventana de edición de individuos incluidos en la base de datos.
3. Abre a la ventana de eliminación de individuos de la base de datos.
4. Amplia la base de datos a partir de las imágenes contenidas en el historial de clasificación. Abre la ventana de comprobación de identidad de los individuos incluidos en el historial.
5. Elimina toda a base de datos.
6. Vuelve al menú principal.

Ventana de actualización y/o inscripción de individuos a la base de datos

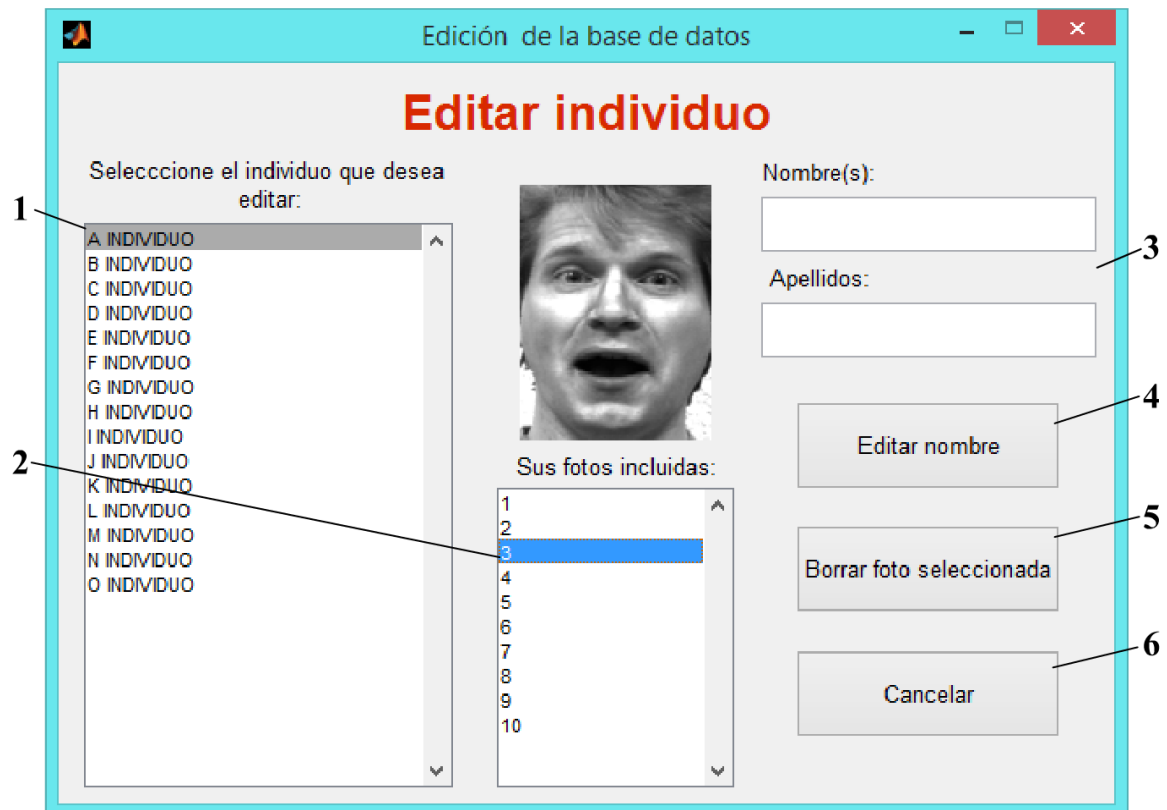
Permite añadir una imagen a la base de datos.



1. Añade una imagen desde archivo. Luego de seleccionar el archivo, si la casilla de verificación “Imágenes precortadas” no está marcada, se abrirá la herramienta de detección y segmentación de imágenes. Posteriormente se abrirá la herramienta de etiquetado de imágenes.
2. Añade varias imágenes contenidas en un directorio. Si la casilla de verificación “Imágenes precortadas” no está marcada, luego de seleccionar el directorio se abrirá la herramienta de detección y segmentación de imágenes para cada imagen contenida en el directorio seleccionado. Posteriormente se abrirá la herramienta de etiquetado de imágenes. Se segmentarán primero **todas** las imágenes y luego se etiquetará cada una.
3. Al marcar esta casilla de verificación se da a entender al programa que las imágenes a agregar han sido segmentadas previamente.
4. Abre la ventana de captura de imagen a través de cámara web. Posteriormente la herramienta de detección y segmentación de rostros, y finalmente la herramienta de etiquetado de imágenes.
5. Vuelve al menú de creación y edición de la base de datos.

Ventana de edición de individuos

Permite cambiar el nombre y apellidos de un individuo, o borrar una de las imágenes asociadas a él.



1. Lista de individuos incluidos en la base de datos.
2. Lista de imágenes incluidas para el individuo seleccionado. Al seleccionar una imagen, ésta podrá visualizarse en el ubicado arriba de la lista.
3. Campos de edición de nombre y apellidos.
4. Confirma el cambio de nombre del individuo seleccionado.
5. Borra la imagen seleccionada.
6. Cancela los cambios y vuelve al menú de creación y edición de la base de datos.

Ventana de eliminación de individuos

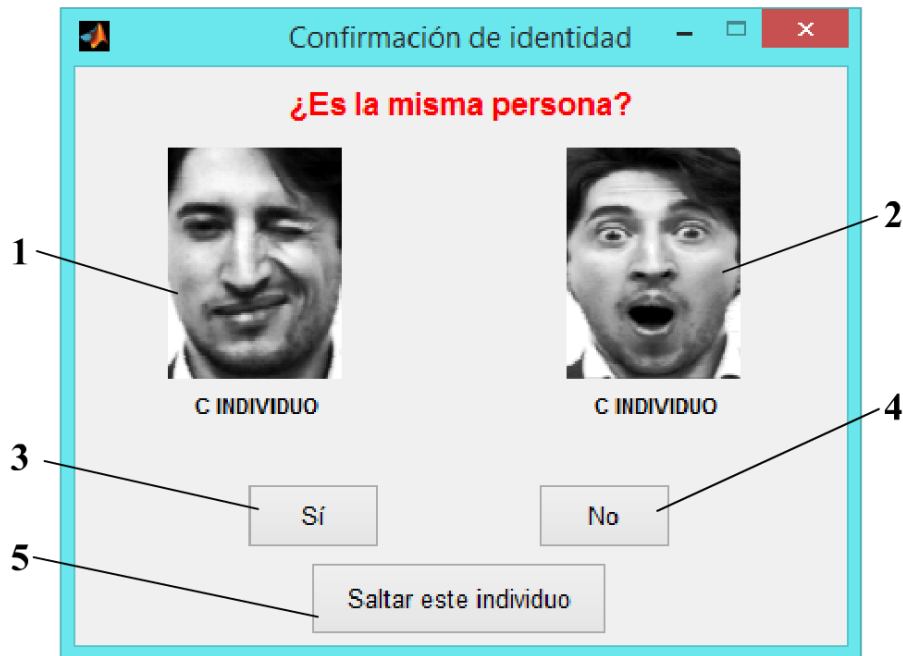
Permite eliminar permanentemente a un individuo de la base de datos.



1. Lista de individuos incluidos en la base de datos.
2. Al seleccionar un individuo de la lista, una de sus imágenes (elegida aleatoriamente) aparecerá en este recuadro. Para ver una distinta, hay que volver a seleccionar el nombre del individuo en la lista.
3. Elimina al individuo seleccionado.
4. Termina la sesión de eliminación de individuos y vuelve al menú de creación y edición de la base de datos.

Ventana de confirmación de identidad

Pide una confirmación sobre las identidades de los individuos incluidos en el historial de identificación. Una vez abierta esta ventana se debe añadir a la base de datos y/o desechar cada imagen del historial. En el caso de cerrar deliberadamente la ventana se perderán todas las imágenes incluidas en el historial.



1. Imagen incluida en el historial. Debajo se aparece la identidad con la que fue almacenada.
2. Imagen en la base de datos (elegida de manera aleatoria) de la identidad que se desea confirmar.
3. Confirmación positiva. Añade a la base de datos la imagen a confirmada. Si la identidad a confirmar fue almacenada en el historial como un individuo desconocido, se abrirá la herramienta de etiquetado de imágenes.
4. Respuesta negativa. Abre la herramienta de etiquetado de imágenes para añadir la imagen en cuestión con su identidad correcta a la base de datos.
5. Salta la imagen en cuestión y pasa a la siguiente.

Ventana de captura de imagen a través de cámara web

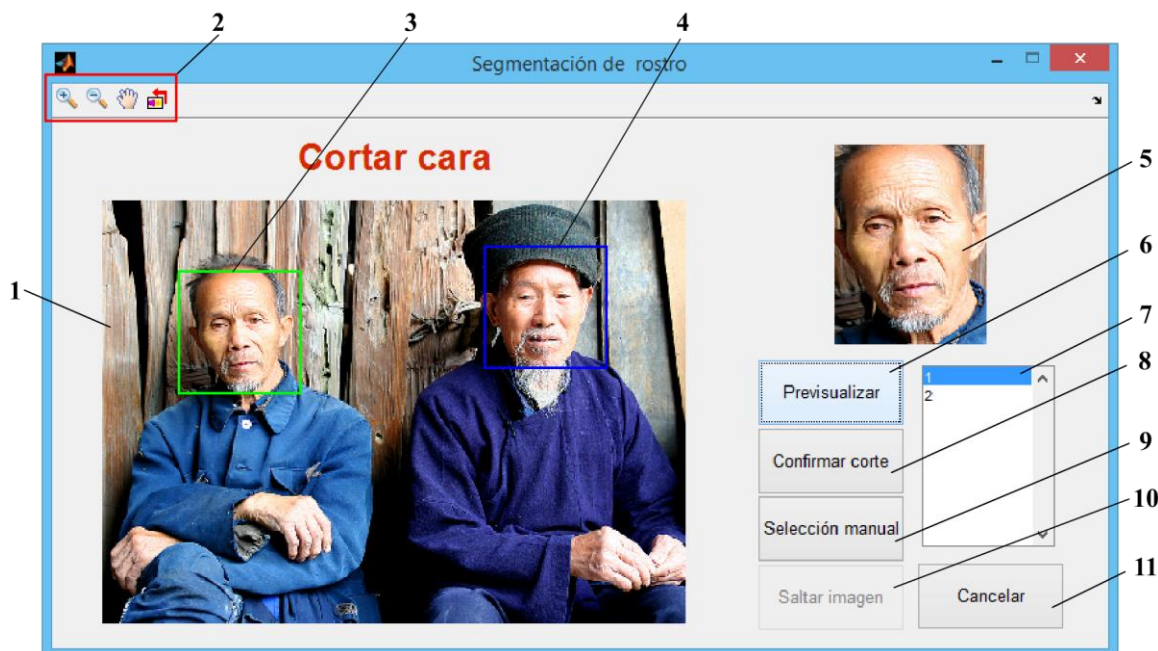
Permite tomar una fotografía.



1. Vídeo capturado por la cámara web.
2. Captura la imagen.
3. Confirma la imagen capturada.
4. Repite la captura.
5. Cancela el proceso de captura de imagen a través de cámara web.

Herramienta de detección y segmentación de individuos (I)

Permite hacer la detección manual o automática de un rostro en una imagen y extraerlo mediante segmentación. Utiliza los porcentajes de corte adicional especificados en las propiedades generales.




1. Imagen a segmentar.
2. Barra de herramientas. En ella se pueden seleccionar las siguientes herramientas:
 - a. Acercar vista (🔍): Acerca la vista de la imagen a segmentar.
 - b. Alejar vista (🔍): Aleja la vista de la imagen a segmentar.
 - c. Arrastre (👉): Arrastra la imagen a segmentar cuando la vista está acercada.
 - d. Girar imagen (🔄): Gira la imagen a segmentar 90°.
3. Rostro seleccionado de la detección automática (recuadro verde).
4. Rostro(s) sin seleccionar de la detección automática (recuadro azules).
5. Marco de previsualización de la segmentación.
6. Actualiza el marco de previsualización.
7. Lista de rostros detectados.
8. Confirma la segmentación.
9. Cambia al modo manual de selección de rostros.
10. Salta la imagen a segmentar y pasa a la siguiente en el caso de que las imágenes hayan sido añadidas por carpeta.
11. Cancela todo el proceso de detección y segmentación.

Herramienta de detección y segmentación de individuos (II)

En el caso de que se haya elegido cambiar al modo manual de selección de rostros o no se disponga del *Computer Vision System Toolbox* de MATLAB, la herramienta de detección y segmentación de individuos tendrá la siguiente apariencia:



1. Nueva herramienta para continuar con la selección () en la barra de herramientas. Con ella se permite continuar con la selección del rostro en el caso de haber utilizado alguna de las otras herramientas.
2. Rectángulo de selección manual. Puede ser ampliado y desplazado.

Herramienta de etiquetado de imágenes

Permite etiquetar una imagen con una nueva identidad, o con la de un individuo ya incluido en la base de datos. Dos individuos pueden compartir el mismo nombre y apellidos, por lo tanto si la imagen a agregar pertenece a un individuo ya incluido en la base de datos, éste debe ser seleccionado en lugar de volver a escribir su nombre.



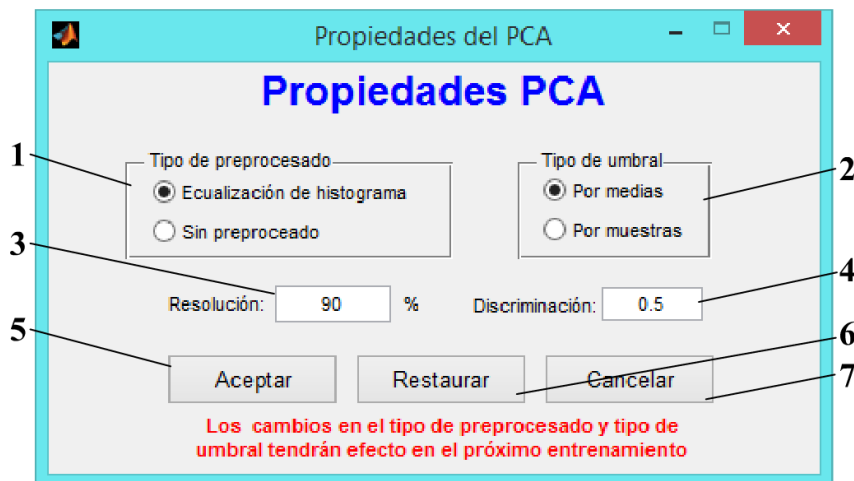
1. Panel de etiqueta nueva. Tiene prioridad sobre la lista de selección de individuos ya incluidos, por lo que los campos de nombre y apellidos deben estar vacíos si la imagen pertenece a un individuo ya incluido en la base de datos.
2. Lista de individuos ya incluidos en la base de datos.
3. Imagen (escogida aleatoriamente) del individuo seleccionado. Para cambiarla debe seleccionarse nuevamente el mismo individuo en la base de datos.
4. Imagen a etiquetar.
5. Etiqueta la imagen con la identidad nueva en caso de estar llenos AMBOS campos de nombre y apellidos. En caso de estar vacíos, etiqueta la imagen con la identidad seleccionada.
6. Cancela el proceso de etiquetado.

Menú del sistema de reconocimiento facial basado en PCA



1. Accede a la ventana de clasificación en modo identificación.
2. Accede a la ventana de clasificación en modo verificación.
3. Entrena el sistema de reconocimiento facial basado en PCA.
4. Abre la ventana de propiedades del sistema basado en PCA.
5. Abre la ventana de visualización del sistema basado en PCA.
6. Vuelve al menú principal.

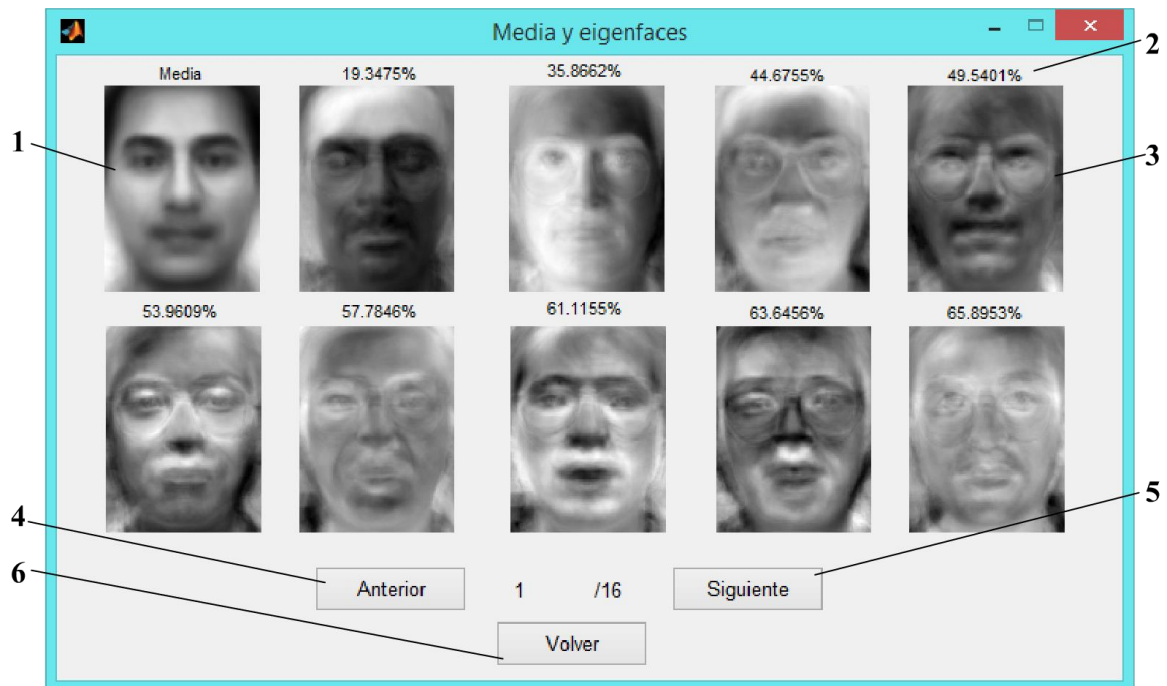
Ventana de propiedades del sistema basado en PCA



1. Panel de elección del preprocesado.
2. Panel de elección del tipo de umbral.
3. Resolución (porcentaje de varianza total representada).
4. Factor de discriminación del sistema.
5. Confirma los cambios realizados.
6. Restaura las propiedades por defecto.
7. Cierra la ventana sin guardar cambios.

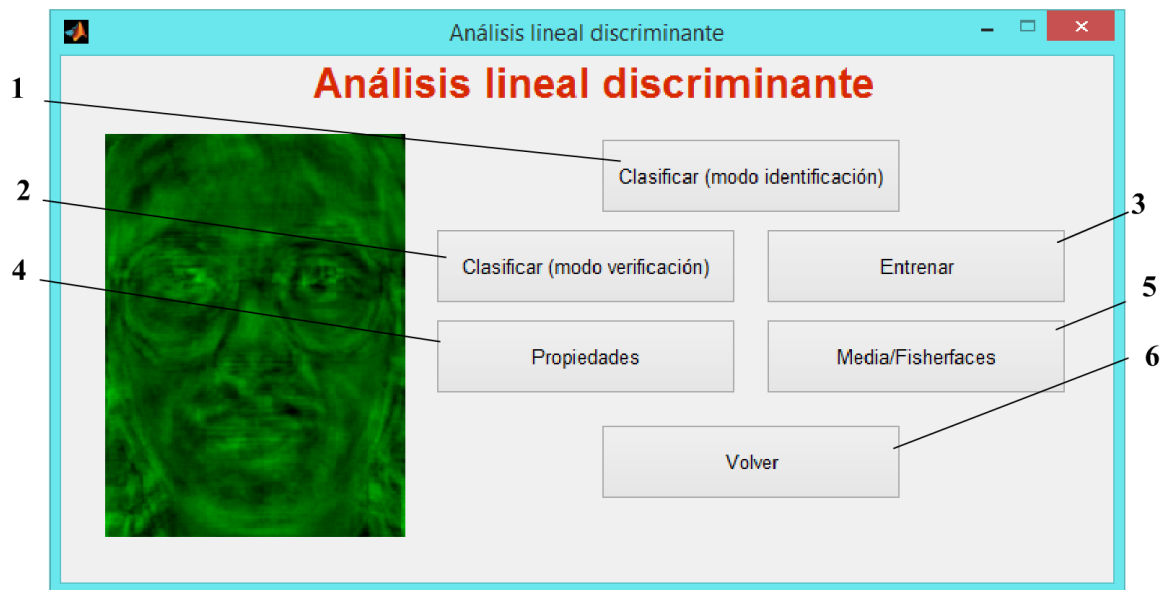
Ventana de visualización del sistema basado en PCA

En ella se observan la media, las *eigenfaces* calculadas, y los porcentajes de varianza total acumulada de cada una de ellas.



1. Media.
2. Porcentajes de varianza total acumulada.
3. *Eigenfaces*.
4. Pasa al anterior conjunto de *eigenfaces* calculadas.
5. Pasa al siguiente conjunto de *eigenfaces* calculadas.
6. Cierra la ventana de visualización.

Menú del sistema de reconocimiento facial basado en LDA



1. Accede a la ventana de clasificación en modo identificación.
2. Accede a la ventana de clasificación en modo verificación.
3. Entrena el sistema de reconocimiento facial basado en LDA.
4. Abre la ventana de propiedades del sistema basado en LDA.
5. Abre la ventana de visualización del sistema basado en LDA.
6. Vuelve al menú principal.

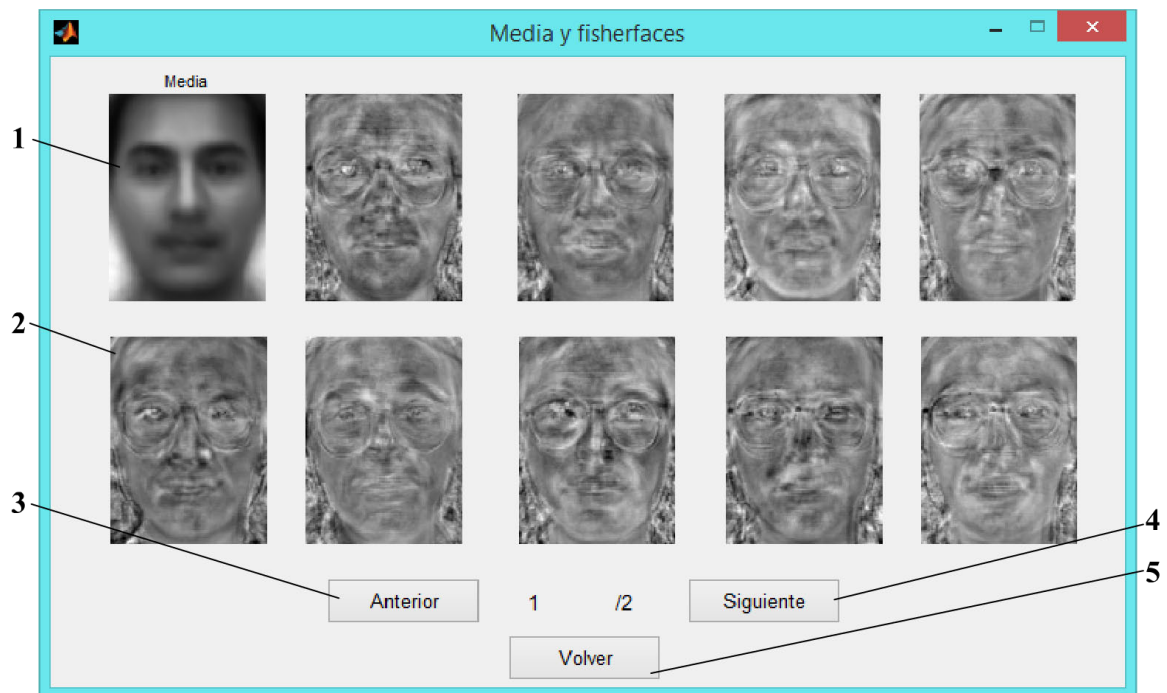
Ventana de propiedades del sistema basado en LDA



1. Panel de elección del preprocesado.
2. Panel de elección del tipo de umbral.
3. Factor de discriminación del sistema.
4. Restaura las propiedades por defecto.
5. Confirma los cambios realizados.
6. Cierra la ventana sin guardar cambios.

Ventana de visualización del sistema basado en LDA

En ella se observan la media y las *fisherfaces* calculadas.



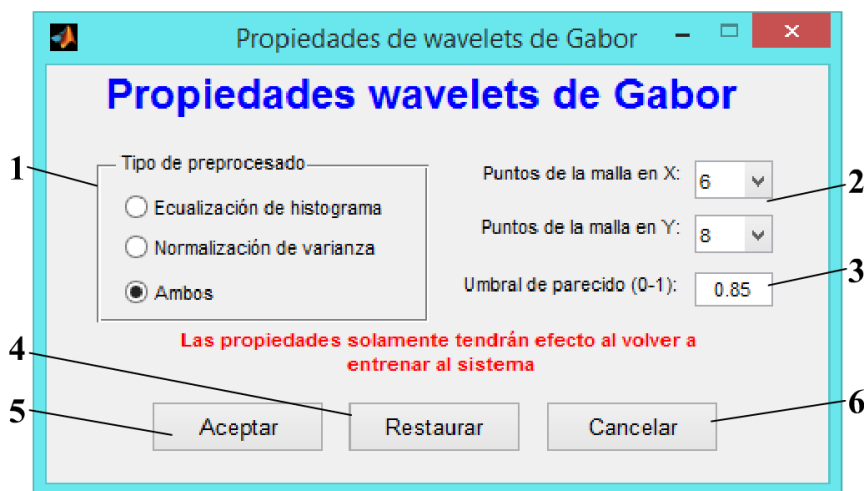
1. Media.
2. *Fisherfaces*.
3. Pasa al anterior conjunto de *fisherfaces* calculadas.
4. Pasa al siguiente conjunto de *fisherfaces* calculadas.
5. Cierra la ventana de visualización.

Menú del sistema de reconocimiento facial basado en *wavelets* de Gabor



1. Accede a la ventana de clasificación en modo identificación.
2. Accede a la ventana de clasificación en modo verificación.
3. Entrena el sistema de reconocimiento facial basado en *wavelets* de Gabor.
4. Abre la ventana de propiedades del sistema basado en *wavelets* de Gabor.
5. Abre la ventana de visualización del sistema basado en *wavelets* de Gabor.
6. Vuelve al menú principal.

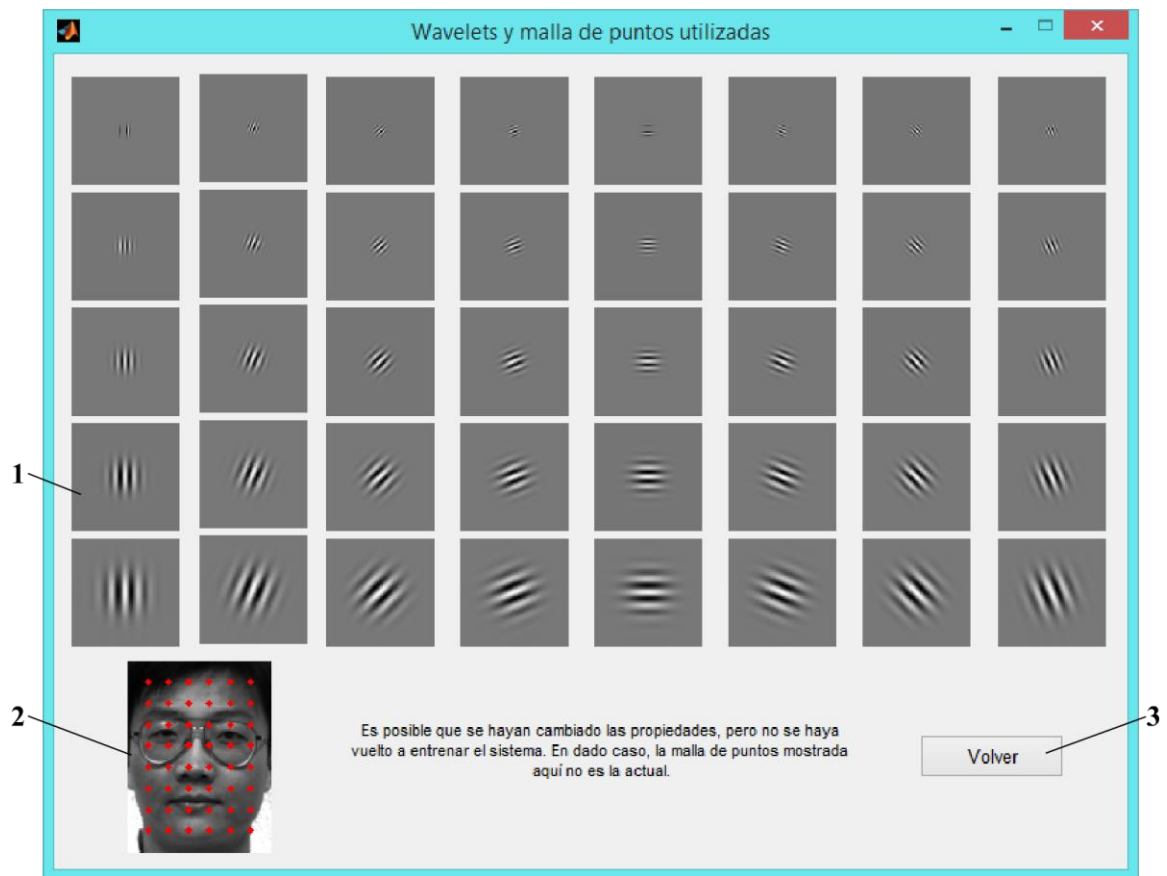
Ventana de propiedades del sistema basado en *wavelets* de Gabor



1. Panel de elección del preprocesado.
2. Selección de puntos equidistantes en cada eje de la malla.
3. Umbral de parecido del sistema.
4. Restaura las propiedades por defecto.
5. Confirma los cambios realizados.
6. Cierra la ventana sin guardar cambios.

Ventana de visualización del sistema basado en *wavelets* de Gabor

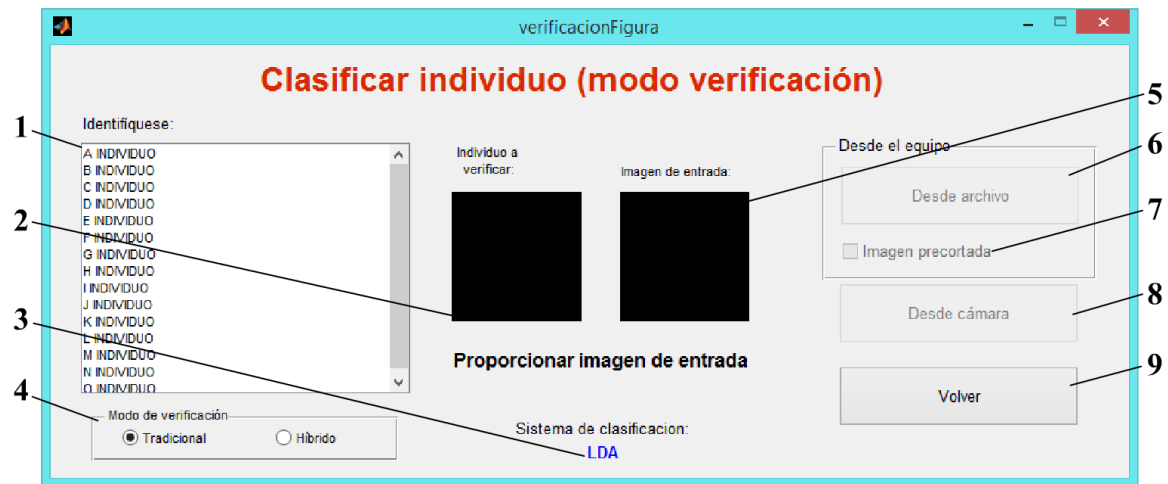
En ella se observan las *wavelets* de Gabor utilizadas y la malla de puntos actual.



1. *Wavelets* de Gabor utilizadas.
2. Malla de puntos actual.
3. Cierra la ventana de visualización.

Ventana de clasificación en modo verificación

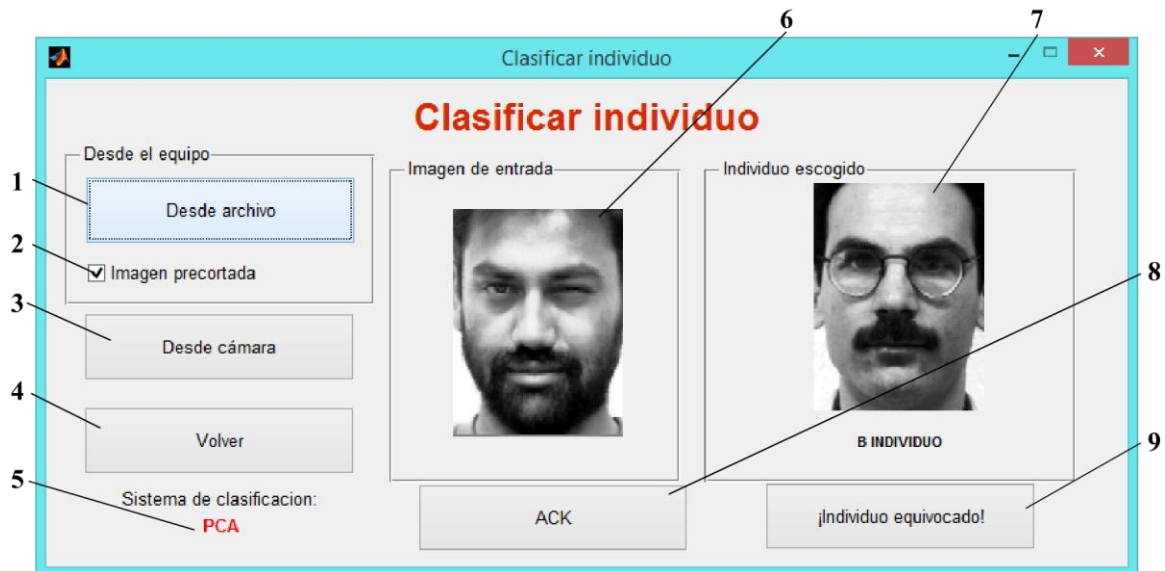
Es la misma ventana para todos los sistemas de reconocimiento facial que ofrecen el modo verificación (PCA, LDA y *wavelets* de Gabor).



1. Lista de individuos incluidos en la base de datos para identificarse.
2. Marco de visualización del individuo seleccionado. Muestra una imagen aleatoria de entre las que contiene dicho individuo en la base de datos. Para cambiarla debe volver a seleccionarse el individuo en la lista.
3. Indicación del sistema utilizado.
4. Selección del modo de verificación a utilizar para la clasificación.
5. Imagen de entrada proporcionada.
6. Clasifica una imagen de entrada desde archivo.
7. Marcando esta casilla de verificación se indica que la imagen de entrada proporcionada desde archivo ha sido previamente segmentada.
8. Clasifica una imagen de entrada desde cámara web.
9. Vuelve al menú del sistema de clasificación desde el cual se accedió a esta ventana.

Ventana de clasificación en modo identificación (I)

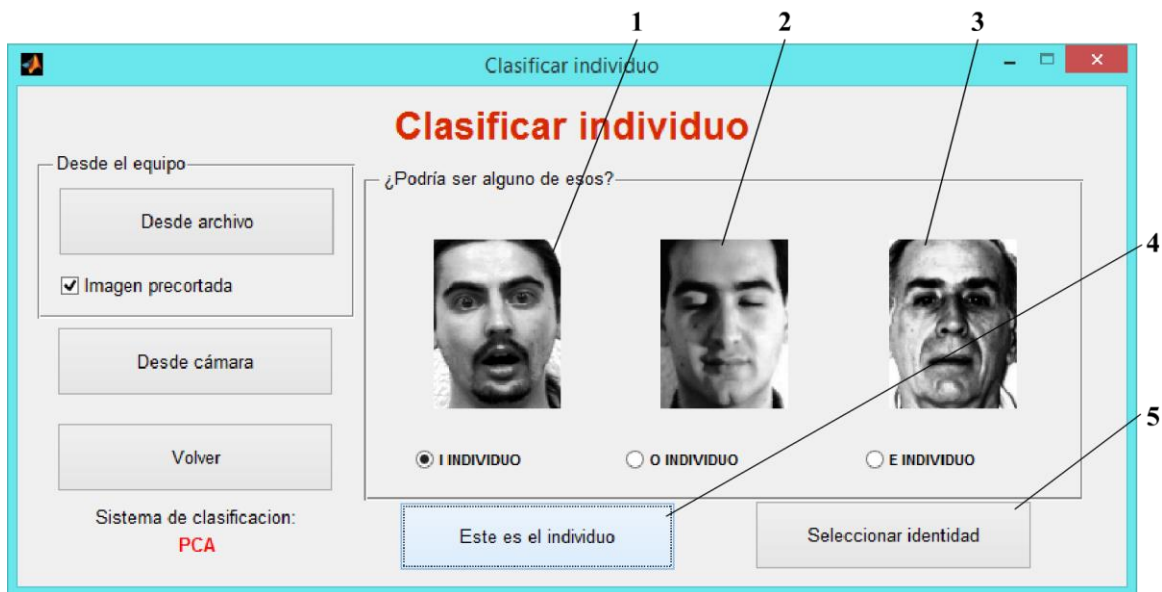
Es la misma ventana para todos los sistemas de reconocimiento facial. Para el sistema basado en EGM se incluye además una casilla de verificación que cuando es marcada muestra el grafo deformado y el grafo original sobre las imágenes.



1. Clasifica una imagen de entrada desde archivo.
2. Cuando esta casilla de verificación está marcada se da a entender al programa que la imagen a clasificar ha sido segmentada previamente.
3. Captura la imagen a clasificar desde cámara web.
4. Vuelve al menú del sistema de clasificación desde el que se accedió a esta ventana.
5. Indicación del sistema de clasificación utilizado.
6. Imagen de entrada.
7. Primera posición de la lista de coincidencias.
8. Confirma que la primera posición de la lista es correcta e incluye la imagen de entrada en el historial de clasificación.
9. Indica que el sistema se equivocó y muestra las siguientes 3 opciones de la lista de coincidencias. Por ejemplo, en la imagen que aquí se muestra se utilizó una resolución de PCA muy baja, con lo cual la primera posición de la lista de coincidencias es incorrecta.

Ventana de clasificación en modo identificación (II)

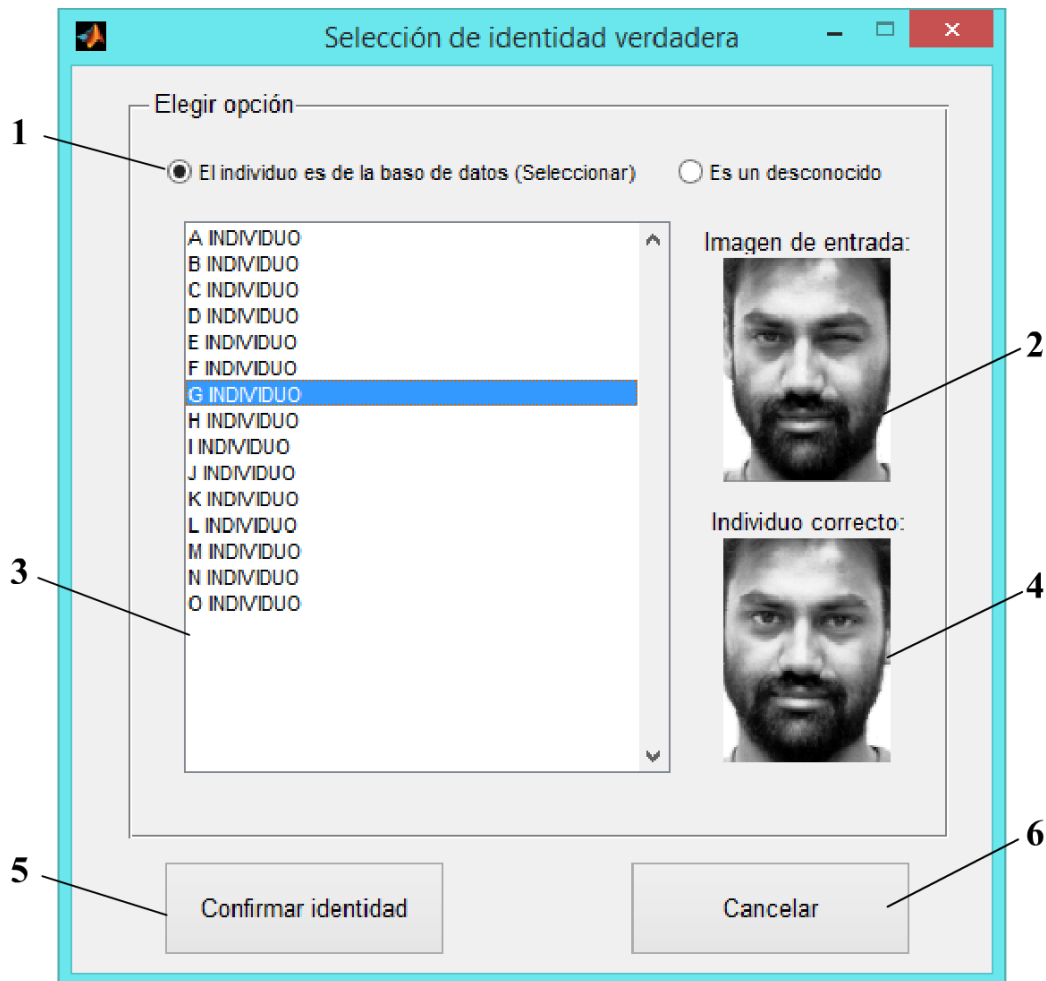
Cuando la primera posición de la lista de coincidencias es incorrecta, debe indicarse la identidad correcta si ésta se encuentra entre las siguientes 3 posiciones de la lista de coincidencias. En consecuencia, la ventana de clasificación en modo identificación cambia su apariencia:



1. Segunda posición de la lista de coincidencias.
2. Tercera posición de la lista de coincidencias.
3. Cuarta posición de la lista de coincidencias.
4. Confirma que la posición seleccionada es correcta e incluye la imagen de entrada en el historial de clasificación.
5. Indica que la identidad correcta no está tampoco en ninguna de las posiciones mostradas. Abre la ventana de identificación fallida para seleccionar la identidad correcta.

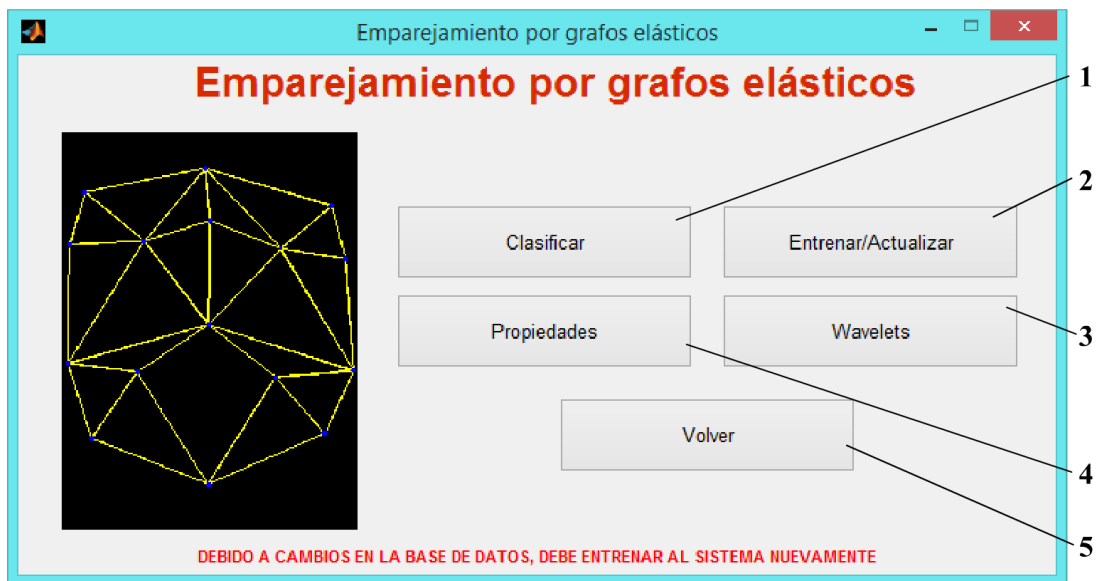
Ventana de identificación fallida

En ella se selecciona la identidad correcta de una imagen de entrada para la cual el sistema de reconocimiento en modo identificación falló.



1. Panel de selección conocido/desconocido.
2. Imagen de entrada que fue clasificada incorrectamente por el sistema.
3. Lista de selección de la identidad correcta de entre los individuos incluidos en la base de datos.
4. Marco de visualización de la imagen del individuo seleccionado. Se escoge aleatoriamente una de las que tiene incluidas en la base de datos. Para mostrar otra imagen debe volver a seleccionarse el mismo individuo en la lista.
5. Confirma la identidad seleccionada como la identidad correcta de la imagen de entrada.

Menú del sistema de reconocimiento facial basado en EGM



1. Accede a la ventana de clasificación en modo identificación.
2. Accede al gestor de modelos.
3. Abre la ventana de propiedades del sistema basado en EGM.
4. Abre la ventana de visualización del sistema basado en EGM.
5. Vuelve al menú principal.

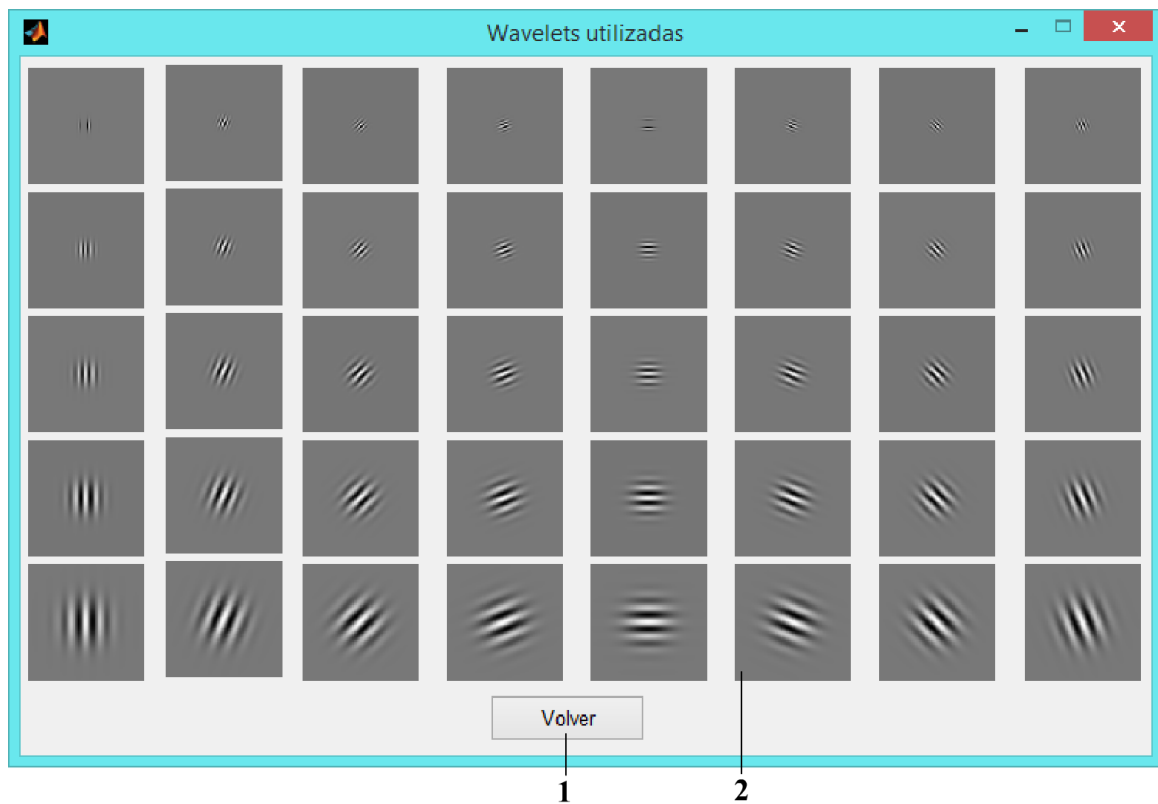
Ventana de propiedades del sistema basado en EGM



1. Panel de elección del preprocesado.
2. Desplazamiento del grafo rígido a lo largo de la imagen en la fase de colocación.
3. Factor de elasticidad.
4. Iteraciones máximas permitidas en la fase de elasticidad sin mejorar el coste.
5. Restaura las propiedades por defecto.
6. Confirma los cambios realizados.
7. Cierra la ventana sin guardar cambios.

Ventana de visualización del sistema basado en EGM

En ella se observan las *wavelets* de Gabor utilizadas.



1. *Wavelets* de Gabor utilizadas.
2. Cierra la ventana de visualización.

Gestor de modelos

A través de este gestor es posible visualizar los modelos creados para los individuos de la base de datos. Además, también se puede acceder a las herramientas de creación de grafos y de creación de modelos.

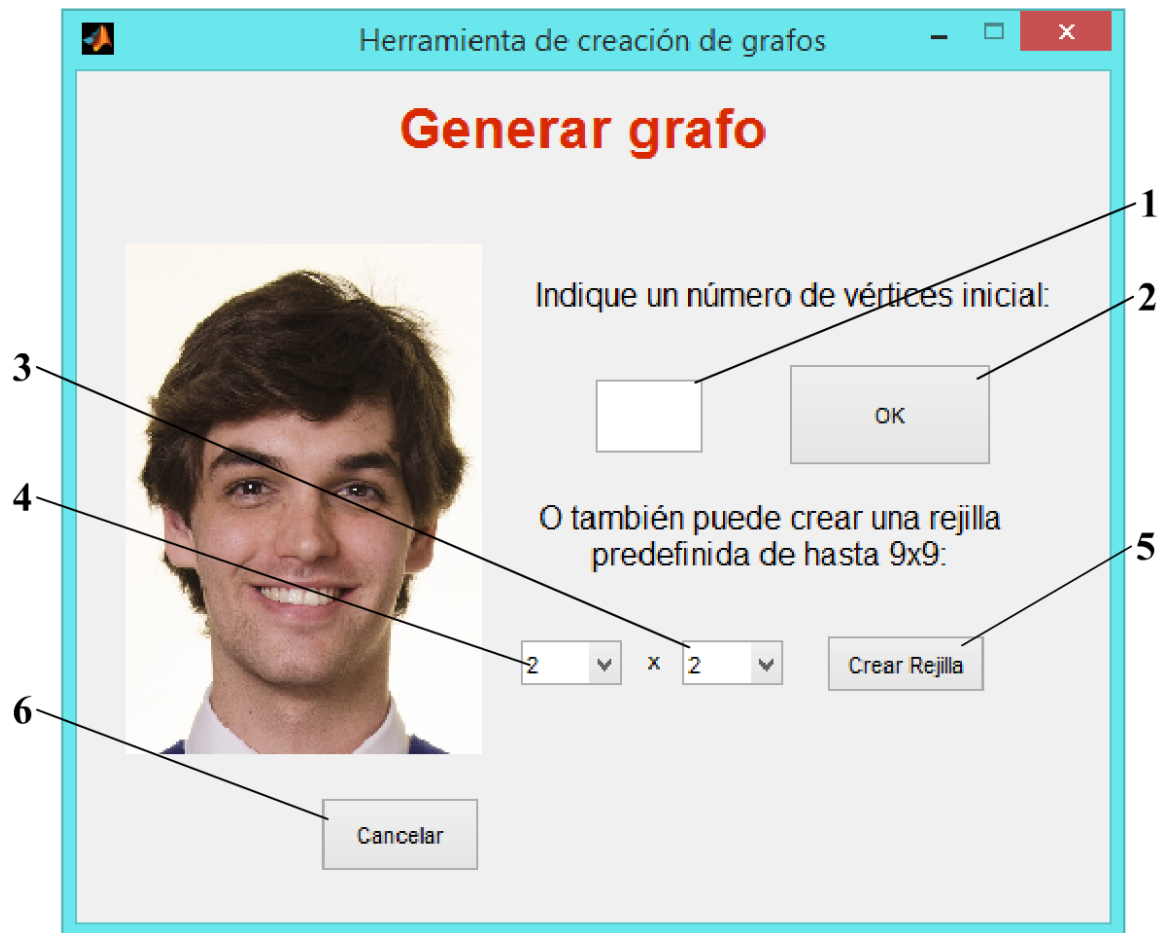
Cuando un individuo de la base de datos tiene asignado un modelo, aparecen tres asteriscos (***) al principio de su nombre.



1. Modelo del individuo seleccionado superpuesto sobre la imagen con la que se creó.
2. Imagen seleccionada del individuo seleccionado.
3. Grafo actual
4. Lista de individuos incluidos en la base de datos.
5. Lista de imágenes del individuo seleccionado
6. Genera el modelo de un individuo a partir de la imagen seleccionada.
7. Elimina el modelo del individuo seleccionado.
8. Cambia el grafo actual. ¡ATENCIÓN! Al cambiar el grafo actual se pierden todos los modelos generados.
9. Vuelve al menú del sistema basado en EGM.

Herramienta de creación de grafos (I)

En ella se permite crear grafos de forma interactiva con la ayuda de una imagen de ejemplo.



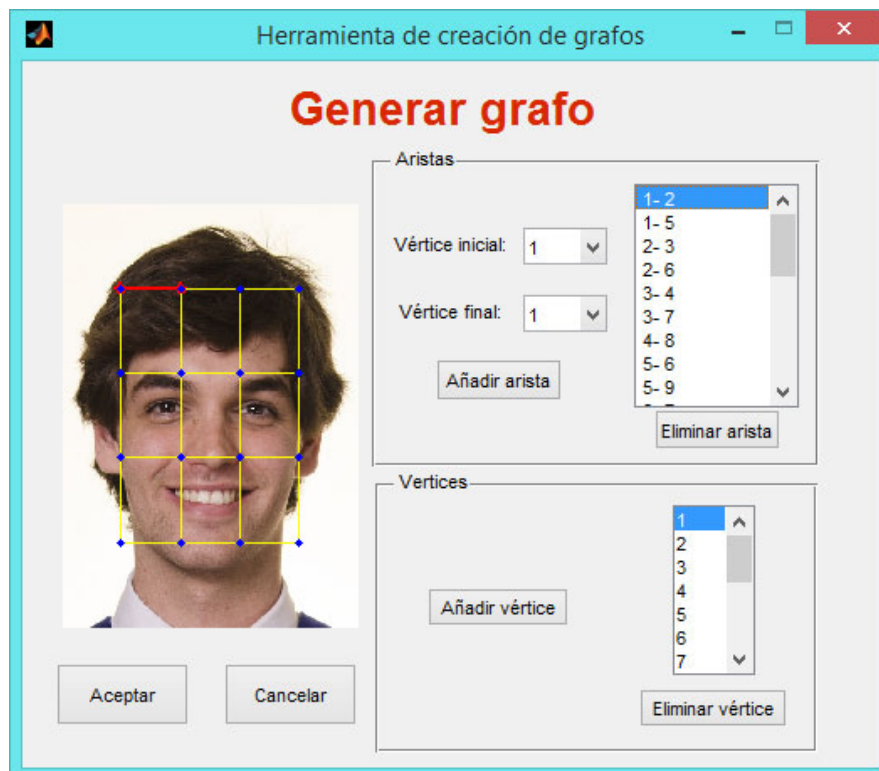
1. Aquí debe indicarse el número de vértices con el que se desea generar el grafo. Luego puede ser editado.
2. Confirmación del número de vértices introducido.
3. Número de vértices equiespaciados a lo largo del eje Y del grafo predefinido.
4. Número de vértices equiespaciados a lo largo del eje X del grafo predefinido.
5. Crea un grafo predefinido (rejilla de vértices conectados con sus vecinos verticales y horizontales).
6. Cancela la creación del grafo.

Herramienta de creación de grafos (II)

Si se opta por indicar el número de vértices iniciales, estos deben ser ubicados en la imagen de ejemplo:

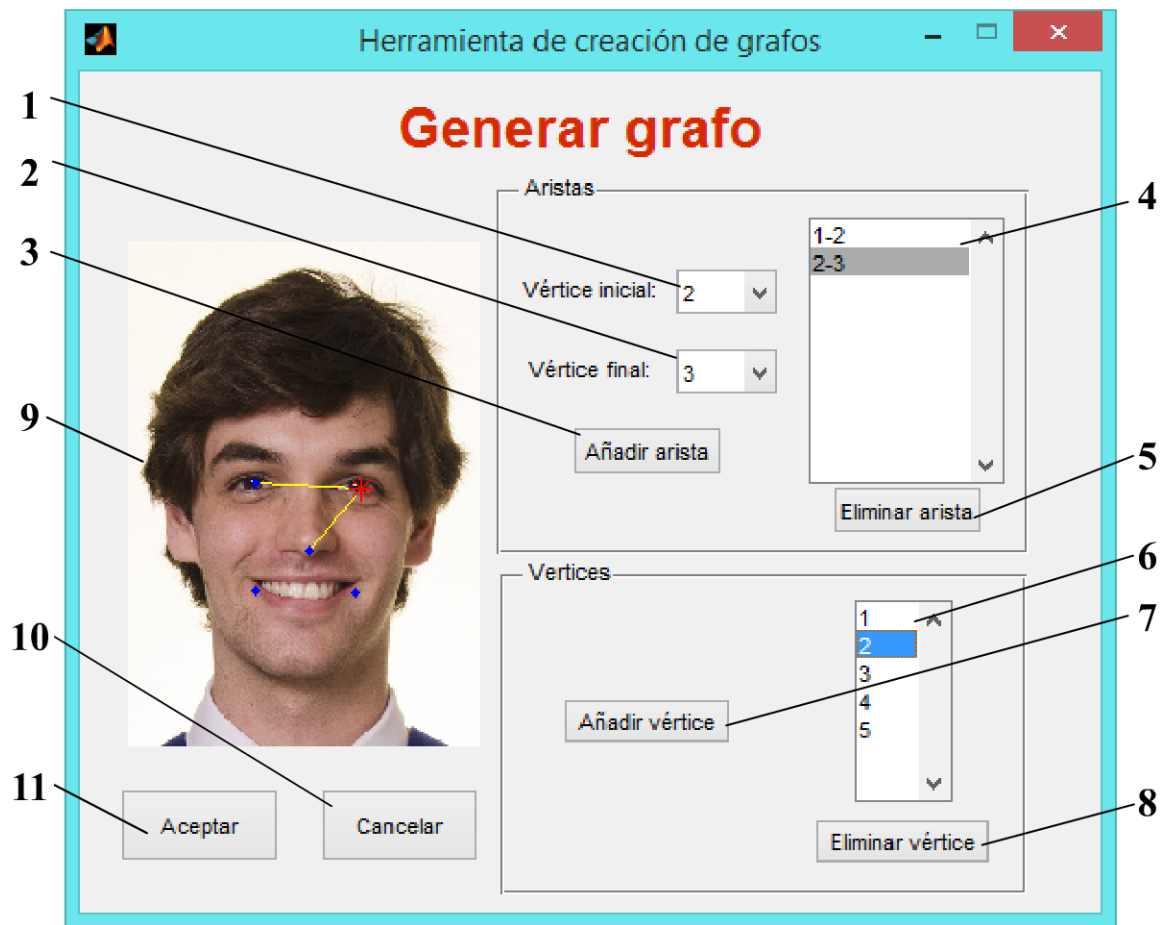


Si se opta por generar un grafo predefinido, aparecerán directamente la opciones de edición del grafo:



Herramienta de creación de grafos (III)

Una vez escogida la apariencia inicial del grafo, éste puede ser modificado:



1. Vértice inicial de la arista que desee añadir.
2. Vértice final de la arista que desee añadir.
3. Añade una arista entre los vértices elegidos.
4. Lista de aristas que contiene el grafo. Cuando una arista es seleccionada, ésta es remarcada con color rojo en la imagen de ejemplo.
5. Elimina la arista seleccionada de la lista.
6. Lista de vértices que contiene el grafo. Cuando un vértice es seleccionado, éste es remarcado con color rojo en la imagen de ejemplo.
7. Añade un vértice al grafo.
8. Elimina el vértice seleccionado.
9. Imagen de ejemplo.
10. Confirma el grafo creado.
11. Cancela el proceso de creación del grafo.

Herramienta de creación de modelos a partir del grafo generado (I)

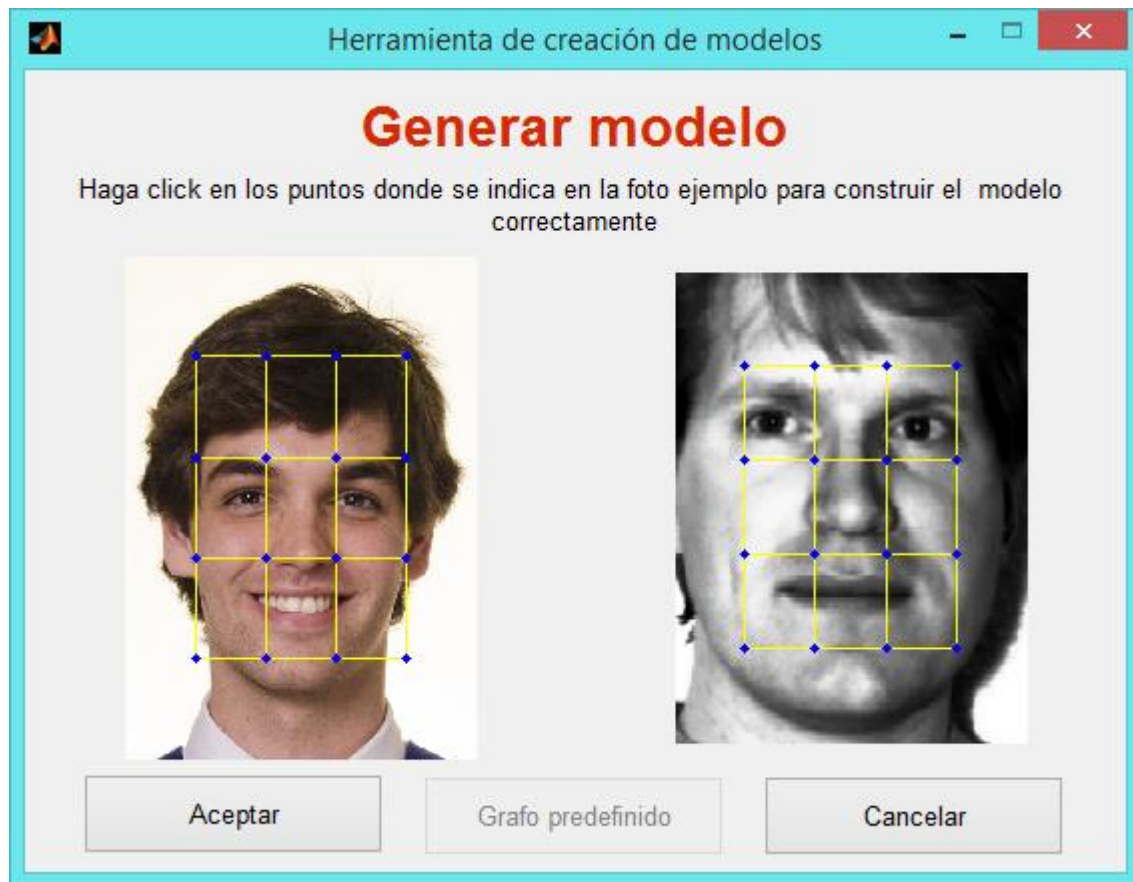
Se muestra una ventana donde por un lado reproduce en la imagen de ejemplo el grafo generado vértice a vértice, y por el otro se pide al usuario que añada (sobre la imagen para la cual se desee crear el modelo) los vértices en las posiciones indicadas en la imagen de ejemplo.



1. Imagen de ejemplo.
2. Posición donde debe ubicarse el siguiente vértice en la imagen para la cual se desee crear el modelo.
3. Imagen para la cual se desea crear el modelo.
4. Cursor.
5. Repite el procedimiento. Esta opción se habilita al final cuando el grafo se reprodujo completamente sobre la imagen en cuestión.
6. Cancela el proceso antes de generar el modelo. Esta opción se habilita al final cuando el grafo se reprodujo completamente sobre la imagen en cuestión.
7. Confirma que los vértices están bien ubicados y genera el modelo. Esta opción se habilita al final cuando el grafo se reprodujo completamente sobre la imagen en cuestión.

Herramienta de creación de modelos a partir del grafo generado (II)

Para el caso en el que se haya creado un grafo predefinido, los vértices serán colocados automáticamente:



Guía de utilización del programa de evaluación

Para iniciar el programa de evaluación, debe seleccionarse el directorio raíz del programa de reconocimiento como el directorio actual y ejecutar la función *evaluar* ya sea abriendo su código fuente y ejecutándola desde la barra de herramientas, o simplemente escribiendo *evaluar* en la ventana de comandos. Una vez hecho esto, se abrirá la ventana principal del programa de evaluación.

Es importante comentar que todos los cambios realizados tanto en las propiedades generales, como en las propiedades de cada sistema evaluado **serán también realizados** en las propiedades del programa de reconocimiento.

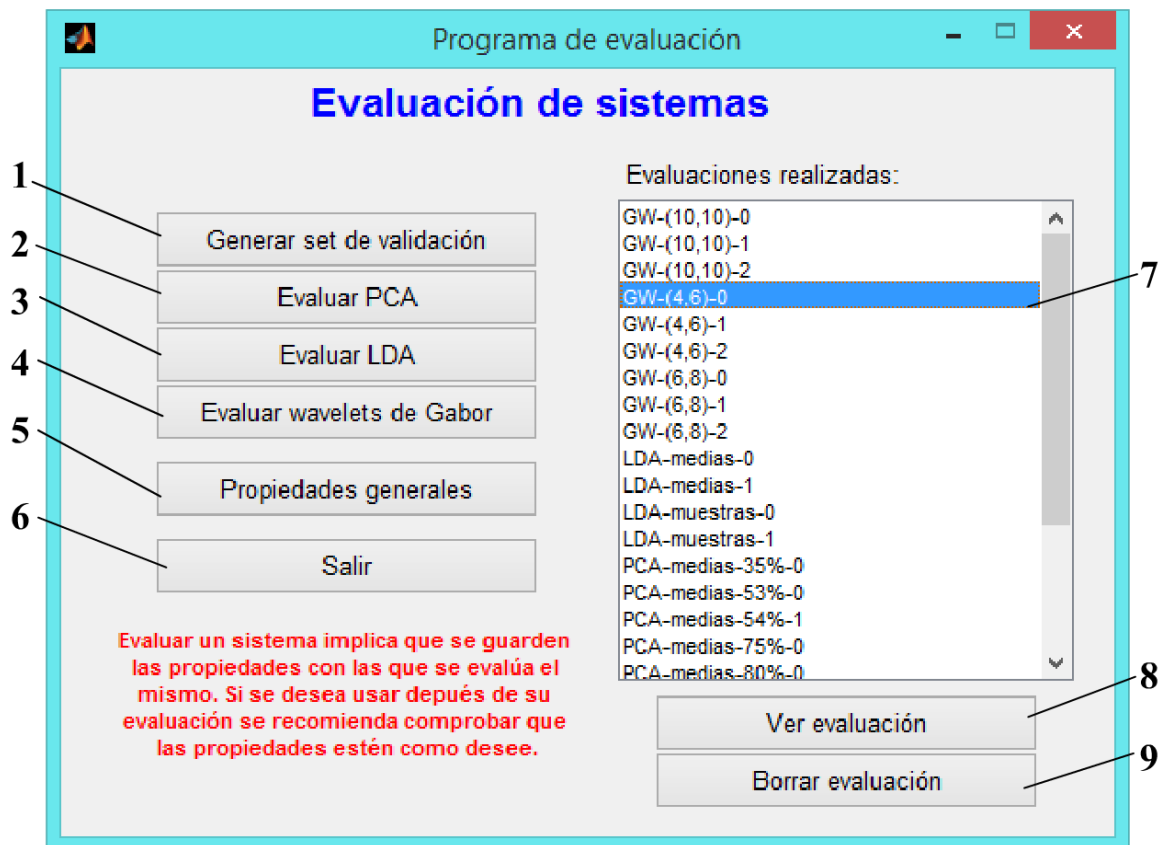
Si es la primera vez que se abre el programa de evaluación, todas las funciones estarán deshabilitadas a excepción de la de generación del set de validación.

Tabla de nomenclatura

El programa de evaluación utiliza la siguiente nomenclatura para referirse a las diferentes configuraciones de los sistemas de reconocimiento:

Técnica	Sistema	Propiedad 1	Propiedad 2	Propiedad 3
Análisis por componentes principales	PCA	medias - Para umbral por medias	Resolución	0 - Para ecualización de histograma
		muestras - Para umbral por muestras		1 - Sin pre-procesado
Análisis lineal discriminante	LDA	medias - Para umbral por medias	0 - Para ecualización de histograma	-
		muestras - Para umbral por muestras	1 - Sin pre-procesado	
Wavelets de Gabor	GW	(PX,PY)	0 - Para ecualización de histograma	-
			1 - Para normalización de varianza	
			2 - Mixto	

Ventana principal del programa de evaluación



1. Accede a la ventana de creación del set de validación.
2. Evalúa el sistema de reconocimiento facial basado en PCA.
3. Evalúa el sistema de reconocimiento facial basado en LDA.
4. Evalúa el sistema de reconocimiento facial basado en *wavelets* de Gabor.
5. Abre la ventana de propiedades generales para cambiar las propiedades de cámara o los porcentajes de corte utilizados en la creación del set de validación.
6. Cierra el programa de evaluación.
7. Lista de evaluaciones realizadas.
8. Visualiza la(s) evaluación(es) seleccionada(s). Puede seleccionarse hasta un máximo de cuatro evaluaciones.
9. Borra la evaluación seleccionada. En caso de haber más de una evaluación seleccionada, solamente se borrará aquella que ocupe la posición más alta de la lista.

Ventana de creación del set de validación

Para cada individuo de la base de datos se pide que se proporcione una imagen de validación. Cuando se añade una imagen que no ha sido segmentada previamente, o se añade la imagen desde cámara web, se abre la herramienta de detección y segmentación del programa de evaluación.

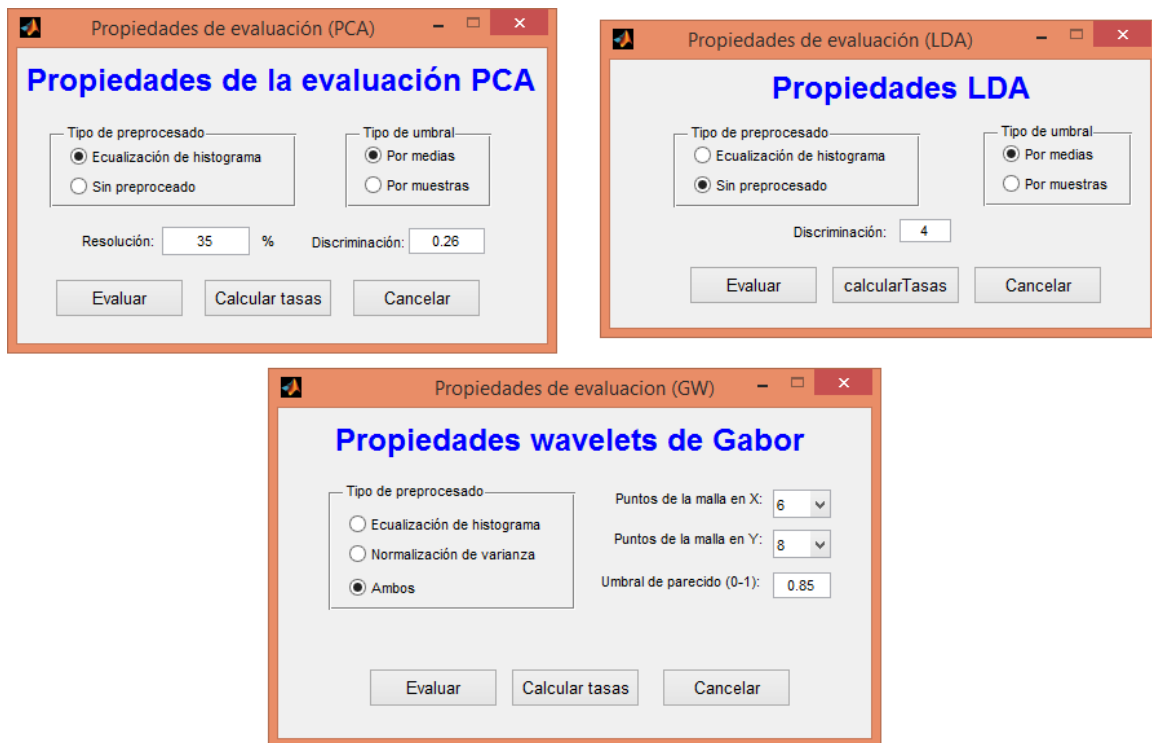


1. Individuo para el cual se pide proporcionar la imagen de validación.
2. Proporciona la imagen de validación desde archivo.
3. Cuando esta casilla de verificación está marcada se da a entender al programa que la imagen a de validación que se va a proporcionar ha sido previamente segmentada.
4. Proporciona la imagen de validación desde captura por cámara web.
5. Cancela el proceso de creación del set de validación.

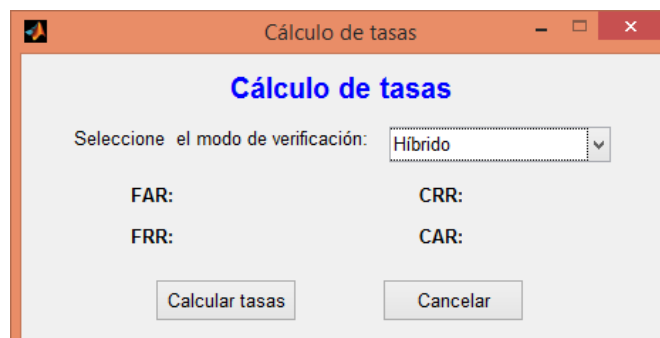
Ventanas de selección de propiedades de evaluación y de cálculo de tasas

Cada vez en el menú principal se indica que se desea evaluar un sistema de reconocimiento, se abre una ventana de selección de propiedades de evaluación. Las propiedades indicadas en estas ventanas forman la configuración con la que será evaluado dicho sistema.

Una vez que el proceso de evaluación termina (se indica mediante barras de espera), la evaluación realizada aparecerá en la lista de evaluaciones del menú principal del programa de evaluación y estará lista para ser visualizada. Las ventanas de propiedades tienen una apariencia parecida a las ventanas de propiedades de cada sistema en el programa de reconocimiento:



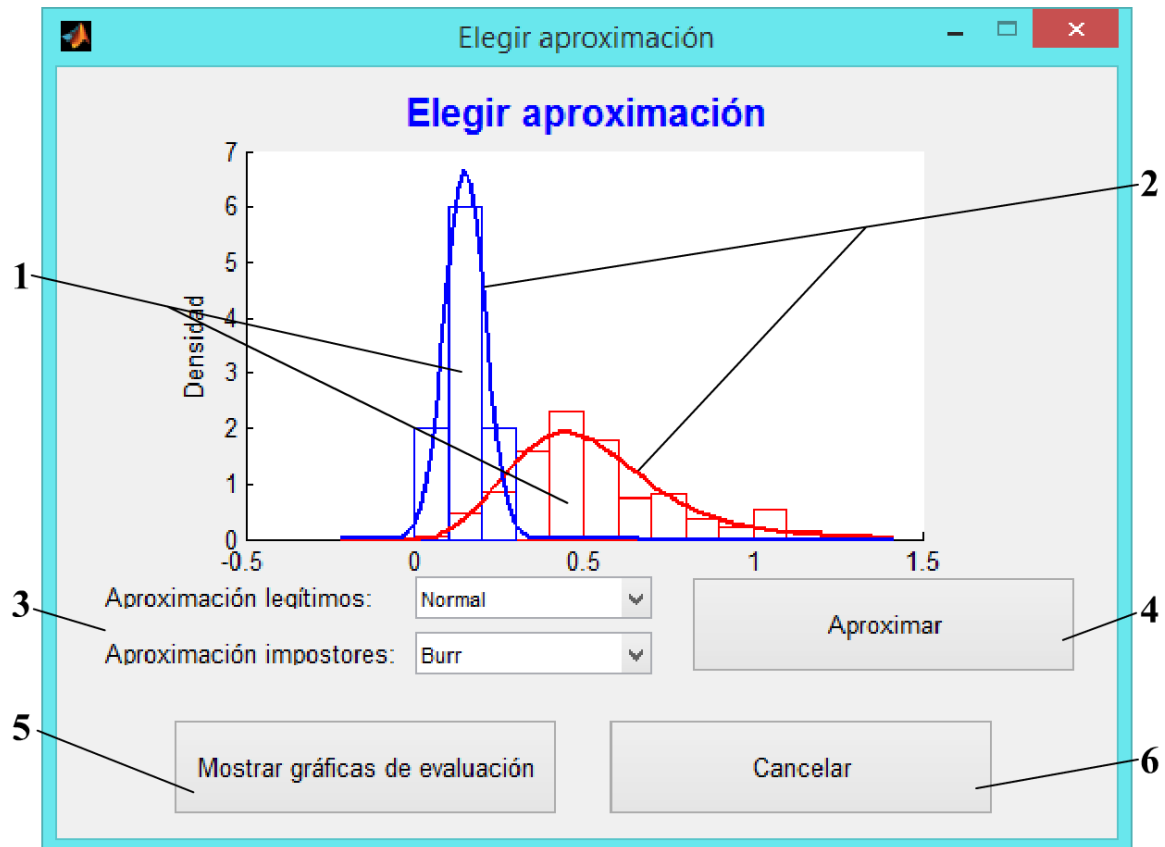
Adicionalmente, en cada ventana de propiedades se puede seleccionar la opción de calcular las tasas de evaluación del sistema para un factor de discriminación/umbral en específico utilizando el modo de verificación tradicional o el híbrido (botón “Calcular tasas”). Cuando se selecciona dicha opción, aparece en pantalla una ventana de cálculo de tasas:



Visualización de las evaluaciones realizadas

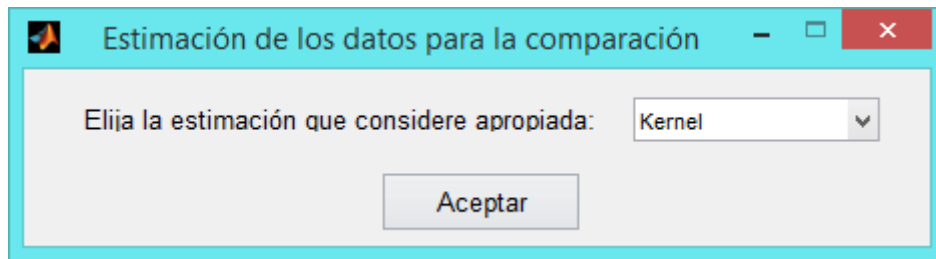
Cuando en el menú principal de evaluación se indica que se desea ver una de las evaluaciones realizadas, se abre una ventana donde se permite elegir la mejor estimación de la distribución de los datos a la distribución estadística conocida que más se desee.

Las gráficas de evaluación mostradas para una evaluación seleccionada son las siguientes: Densidades de probabilidad de individuos legítimos e impostores, curvas de FAR y FRR en función del umbral, ROC y DET.



1. Densidad de probabilidad de los datos empíricos (histogramas).
2. Estimaciones a la distribución estadística escogida (curvas).
3. Selección de las distribuciones estadísticas deseadas. Si son cambiadas, es preciso actualizar la gráfica.
4. Actualiza la gráfica a las distribuciones escogidas.
5. Muestra las curvas de evaluación escogidas.
6. Cancela el proceso de visualización de la evaluación escogida en el menú principal.

Si en el menú principal se indicó que se deseaban ver más de una evaluación, se elegirá la estimación con una ventana más simple, y se mostrarán las curvas ROC de las evaluaciones seleccionadas en una sola gráfica, y las DET en otra:



Tablas de propiedades por defecto

Propiedades generales por defecto

Propiedades de cámara			
Adaptador	<i>winvideo</i> , si existiese. En caso contrario, se deshabilita la captura por cámara web hasta que el usuario la configura en la ventana de propiedades generales.		
Dispositivo	El primero que se detecte conectado al adaptador <i>winvideo</i> . En caso de no haber ninguno, se deshabilita la captura por cámara web.		
Formato	El primero que se detecte que permita el dispositivo por defecto.		
Propiedades de cámara		Propiedades de cámara	
Porcentaje lateral	15	Ancho	192
Porcentaje superior	30	Alto	256
Porcentaje inferior	5	Formato	PGM

Propiedades por defecto del sistema de reconocimiento facial basado en PCA

Propiedades por defecto del sistema de reconocimiento basado en PCA	
Pre-procesado	Ecualización de histograma
Tipo de umbral	Por medias
Factor de discriminación	0,5
Resolución	Aquella que asegure una representación del 90% de varianza como mínimo para el último entrenamiento
Resolución máxima	Número de <i>eigenfaces</i> resultantes del último entrenamiento

Propiedades por defecto del sistema de reconocimiento facial basado en LDA

Propiedades por defecto del sistema de reconocimiento basado en LDA	
Pre-procesado	Ecualización de histograma
Tipo de umbral	Por medias
Factor de discriminación	6

Propiedades por defecto del sistema de reconocimiento facial basado en *wavelets* de Gabor

Propiedades por defecto del sistema de reconocimiento basado en <i>wavelets</i> de Gabor	
Pre-procesado	Mixto
Umbral	0,85
Puntos de la malla en el eje X	6
Puntos de la malla en el eje Y	8

Propiedades por defecto del sistema de reconocimiento facial basado en EGM

Propiedades por defecto del sistema de reconocimiento basado EGM	
Pre-procesado	Mixto
Desplazamiento de la etapa de colocación	10
Factor de rigidez	0,005
Iteraciones máximas sin reducir el coste de la etapa de elasticidad	20

Referencia citada en este manual de usuario

- [1] Yale University, «Yale faces,» [En línea]. Available: <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>. [Último acceso: diciembre 2013].